

SHELF MANAGEMENT SOFTWARE

SOFTWARE VERSION 4.2.x

Revision history

Version	Date	Description
-0000	December 2006	First edition.
-0001	September 2007	Second edition. New features and editorial corrections. See the <i>Release Notes</i> for new feature descriptions.
-0002	May 2008	Third edition. SNMP access to HPI services. New features related to ATCA software version 3.2.0 and editorial corrections. The command line information has been removed and now resides in the <i>CLI Reference</i> . See the <i>Release Notes</i> for new feature descriptions.
-0003	September 2008	Fourth edition. Support for ATCA-6014 and ATCA-6016 shelves. New sensor, troubleshooting, and alarm information.
-0004	January 2009	Fifth edition. Support for updated PICMG and HPI specifications. New features include user-definition of alarm severities and a new fan floor algorithm.
-0005	July 2009	Sixth edition. Support for ATCA-6002 shelf and additional shelf management-capable modules. Added support for ATCA-5014 SPM and discontinued support for ATCA-5100 SPM.
-0006	August 2010	Seventh edition. Support for updated HPI specification, SAI-HPI-B.03.02. Added support for modular HPI framework and blade management through HPI. Made Shelf Manager enhancements. Added support for ATCA-4500 as Shelf Manager in ATCA-6002 shelf.
-0007	December 2011	Eighth edition. Updated for the ATCA software version 4.2.0. Added support for the ATCA-4550 and the ATCA-4555 CPMs. Removed support for the ATCA-4300 and the ATCA-4310 CPMs. Updated the information on disabling and enabling the Shelf Manager. Made other minor editorial corrections.
-0008	April 2012	Ninth edition. See What's new in this manual on page 6 for a description of changes in this edition.

© 2006–2012 by RadiSys Corporation. All rights reserved.

Radisys is a registered trademark of RadiSys Corporation. AdvancedTCA, ATCA, and PIGMG are registered trademarks of PCI Industrial Computer Manufacturers Group. MontaVista is a registered trademark of MontaVista Software Inc. Red Hat and Red Hat Linux are registered trademarks of Red Hat, Inc. Wind River is a registered trademark of Wind River Systems, Inc. Linux is a registered trademark of Linus Torvalds. All other trademarks, registered trademarks, service marks, and trade names are the property of their respective owners.

Table of Contents

Preface	6
About this manual.....	6
What's new in this manual	6
Shelves and modules supported	7
Where to get more product information	8
Notational conventions	9
Chapter 1: Shelf Management Overview	10
Shelf management features	10
Shelf Manager compliance	12
Physical distribution of the shelf management infrastructure	12
Major elements of the shelf management infrastructure.....	14
Chapter 2: Software Architecture	18
Shelf management architecture.....	18
HPI architecture.....	20
Basic and enhanced shelf management operation.....	24
Software redundancy.....	25
Shelf cooling.....	33
Chapter 3: Configuring Shelf Manager and HPI Behavior	36
Disabling or enabling the Shelf Manager	36
Changing the Shelf Manager operating mode	39
Overview of Shelf Manager and HPI configuration	40
Editing the shmgr.conf configuration file	42
Configuring Shelf Manager behavior	43
Configuring HPI behavior	46
Configuring the Shelf Manager in basic mode.....	50
Editing the rsyshsd.conf configuration file	51
Configuring blade HPI behavior	51

Table of Contents

Chapter 4: Changing Shelf Settings Using HPI	53
Radisys HPI implementation details	53
Using the platform-management CLI	56
Writing HPI applications	57
Using OpenHPI tools	59
Using SNMP	60
Using the example HPI application (hpiapp).....	63
Changing the shelf address	65
Changing the Shelf Manager IP address	66
Configuring the FRU power-on sequence	67
Querying the current hot-swap state of a FRU	71
Overriding the default ShMS hot-swap mechanism.....	72
Overriding the cooling algorithm.....	74
Changing shelf power properties	76
Saving and loading custom FRU configurations	76
Chapter 5: Managing Alarms and Events.....	82
Shelf Manager role in alarm management.....	82
Domain alarm types and troubleshooting overview	83
Domain alarm table entries.....	87
Assigning event severities	87
ATCA-6000 shelf display and alarm indicators.....	92
ATCA-6006 shelf alarm indicators	94
ATCA-6014 and ATCA-6016 shelf alarm indicators	95
Managing shelf alarm indicators.....	95
Chapter 6: Troubleshooting	98
Diagnostic tools	98
General troubleshooting procedures	100
IPMI and IPMB troubleshooting topics.....	104
Troubleshooting FRU and shelf operation issues.....	110
Troubleshooting Shelf Manager issues	121
Resolving shelf FRU information issues	131

Table of Contents

- Appendix A: Shelf Manager Initialization 137**
 - Shelf Manager and HPI interface initialization 137
 - HPI subagent initialization 139
 - Initialization in basic mode 139
- Appendix B: Shelf Management Files 140**
 - Shelf management configuration files 140
 - Shelf Manager log file 141
 - Shelf management executable files 141
- Appendix C: IPMI Commands and Managed Sensors 142**
 - Supported IPMI commands 142
 - Sensors managed by the Shelf Manager 142

Preface

About this manual

This manual describes the Radisys Shelf Management Software (the “Shelf Manager”), the Hardware Platform Interface (HPI) server framework, the Intelligent Platform Management Interface (IPMI) infrastructure, and the related software interfaces for AdvancedTCA® (ATCA®) shelves. The Shelf Manager is a software component that resides on a supported front module. The shelf management software monitors IPMI sensors and controls, responds to hardware events, and promotes smooth operation of ATCA modules and shelf components in a Radisys shelf.

This manual describes the shelf management architecture, operation, and application programming interface. The manual provides usage information for shelf management tools and command interfaces, and assumes you know how to use and configure operating systems and networks.

For command syntax, mode, description and related commands see the *Command Line Interface Reference*.

What’s new in this manual

- Added support for the following modules:
 - ATCA-4616 Compute Processing Module (CPM)
 - ATCA-4618 CPM
 - ATCA-4648 CPM
- Added information about ATCA-45xx series and ATCA-46xx series CPMs operating in enhanced mode by default (see [page 24](#)).
- Updated the list of menu options for the hpiapp (see [page 64](#)).

Preface

Shelves and modules supported

The Shelf Manager described in this manual is available on these Radisys shelves with a supported front module installed.

Figure 1. Shelf Manager support

Full name of shelf	Referred to in this manual as	Modules supporting the Shelf Manager
ATCA-6000 12U/14-slot shelf	ATCA-6000 shelf	ATCA-2210 Switch and Control Module (SCM)
ATCA-6006 5U/6-slot shelf	ATCA-6006 shelf	
ATCA-6014 13U/14-slot shelf	ATCA-6014 shelf	
ATCA-6016 13U/16-slot shelf	ATCA-6016 shelf	
ATCA-6002 2U/2-slot shelf	ATCA-6002 shelf	ATCA-1200 Quad AMC Carrier ATCA-45xx series CPMs: <ul style="list-style-type: none">• ATCA-4500• ATCA-4550• ATCA-4555 ATCA-46xx series CPMs: <ul style="list-style-type: none">• ATCA-4616• ATCA-4618• ATCA-4648 ATCA-7220 Packet Processing Module ATCA-9100 Media Resource Module

In the ATCA-6002 shelf, the module supporting the Shelf Manager is installed in slot 1 and can operate in basic or enhanced mode. The default operating mode for most of the modules that support the Shelf Manager, their default operating mode is basic mode. The exception are the ATCA-45xx series and the ATCA-46xx series CPMs, which only operate in the enhanced mode when they are installed in the ATCA-6002 shelf.

Except where noted, the shelf management features described in this manual are supported only in enhanced mode. For details on basic and enhanced modes, refer to [Basic and enhanced shelf management operation on page 24](#).

Throughout this manual, the modules supporting the Shelf Manager will be referred to as “Shelf Manager modules.” If information applies only to a certain module, the specific name of the module will be used, such as “SCM.”

Note: The Shelf Manager and the blade HPI server (introduced in [HPI architecture on page 20](#)) cannot be used when SELinux is installed, because file permission restrictions prevent the Shelf Manager from bringing up its RMCP interfaces.

Preface

Where to get more product information

Visit the Radisys web site at www.radisys.com for product information and other resources. Downloads (manuals, release notes, software, etc.) are available at www.radisys.com/downloads.

Related Radisys manuals

The following software and hardware information is available for Radisys modules:

- **Front module installation and initial setup instructions.** The installation guide for each front module provides the steps for installing the module into a shelf and completing the initial configuration.
- **Front module hardware and troubleshooting information.** The reference manual for each module describes the module's hardware features and operation. The reference manuals also contain information on Intelligent Platform Management Interface (IPMI) commands, diagnostic tests, and troubleshooting.
- **Platform hardware information.** The reference manual for each Radisys system describes the platform and its primary hardware component, the shelf.
- **Command line interface (CLI) reference information.** The *Command Line Interface Reference* describes the Radisys CLI and its command modes and serves as a reference for command syntax and options. Throughout this document, this manual is referred to as the *CLI Reference*.
- **Software reference information.** The *Software Guide for Management Processors and General Computing Processors* describes software concepts and serves as a reference for procedural and usage information. Throughout this document, this manual is referred to as the *Software Guide*.

Sources for standards information

For more information about the PCI Industrial Computer Manufacturers Group (PICMG®) and the AdvancedTCA standard, consult the PICMG Web site at this URL: <http://www.picmg.org>

For more information about the Service Availability Forum (SAF) and the HPI standard, consult the SAF Web site at this URL: <http://www.saforum.org>

For more information about IPMI and the IPMI standards, consult the Intel Web site at this URL: <http://developer.intel.com/design/servers/ipmi>

Specifications and standards documents

IPMI Intelligent Platform Management Bus Communications Protocol Specification, v1.0, Document Revision 1.0, Intel, Hewlett-Packard, NEC, Dell, November 15, 1999.

IPMI Platform Management FRU Information Storage Definition, v1.0, Document Revision 1.1, Intel, Hewlett-Packard, NEC, Dell, September 27, 1999.

Preface

IPMI Intelligent Platform Management Interface Specification Second Generation, v2.0, Document Revision 1.0, Intel, Hewlett-Packard, NEC, Dell, February 12, 2004, 2/15/06 Markup.

PICMG 3.0 Revision 3.0 AdvancedTCA Base Specification, March 24, 2008, PICMG.
(Throughout this manual, this specification is referred to as the *AdvancedTCA Base Specification*.)

PICMG 3.1 Revision 1.0 Specification, Ethernet/Fibre Channel for AdvancedTCA Systems, January 22, 2003, PICMG.

PICMG AMC.0 R1.0 Advanced Mezzanine Card Base Specification, ECN001, June 22, 2006, PICMG.

PICMG HPM.1 Hardware Platform Management IPM Controller Firmware Upgrade Specification, PICMG HPM.1 Draft 0.8, November 20, 2006, PICMG.

Service Availability Forum Hardware Platform Interface Specification, SAI-HPI-B.03.02, August 4, 2009. (Throughout this manual, this specification is referred to as the *SAF HPI Specification*.)

Service Availability Forum Application Interface Specification, SAI-AIS-A.01.01, April 12, 2003.

Service Availability Forum HPI-to-AdvancedTCA Mapping Specification, SAIM-HPI-B.01.01-ATCA, December 20, 2005. (Throughout this manual, this specification is referred to as the *SAF Mapping Specification*.)

Notational conventions

This manual uses the following conventions:

<i>ItalicText</i>	File, function, and utility names.
MonoText	Screen text and syntax strings.
BoldMonoText	A command to enter.
<i>ItalicMonoText</i>	Variable parameters.
Brackets []	Command options.
Curly braces { }	A grouped list of parameters.
Vertical line	An “OR” in the syntax. Indicates a choice of parameters.

All numbers are decimal unless otherwise stated. Bit 0 is the low-order bit. If a bit is set to 1, the associated description is true unless otherwise stated.

Shelf Management Overview

The Shelf Manager is responsible for monitoring conditions of modules and other shelf components and controlling their operation in order to keep them working properly. The Shelf Manager works together with the Intelligent Platform Management Interface (IPMI) infrastructure to monitor the status of the system and correct problems when necessary. The Shelf Manager reports events and anomalies to a system manager and responds to action requests from the system manager.

The hardware platform interface (HPI) provides a standard application programming interface (API) for system manager development.

Shelf management features

The Shelf Manager automatically starts at boot time and is required to be fully operational before the rest of the platform. In conjunction with the other shelf management components, the Shelf Manager monitors platform health, manages events and error conditions, and provides management interfaces for remote interactions.

Platform health monitoring

The Shelf Manager performs the following operations to monitor platform health:

- Gathers and tracks the shelf inventory and associated resource data records (RDRs) in the resource presence table (RPT), including each sensor data record (SDR)
- Retrieves sensor data
- Logs system events
- Receives failure notifications and other events from intelligent FRUs in the shelf
- Responds to alarms with audible or visible indications or event messages to the system manager as appropriate

Event and error management

The Shelf Manager manages normal events and error conditions in these ways:

- Powers devices up and down as they are inserted and removed
- Confirms module compatibility using electronic keying before enabling backplane ports for an inserted module
- Manages power by negotiating the power to allocate for each FRU
- Manages temperatures, adjusting fan speeds as necessary
- Recovers from failures by resetting or power cycling a FRU if warranted
- Responds to remote management requests
- Coordinates with a redundant Shelf Manager on a redundant SCM to negotiate which Shelf Manager is active and to synchronize data so the standby Shelf Manager is ready for activation if needed

Interfaces for shelf management

The Shelf Manager provides the following management interfaces:

- HPI through Remote Management Control Protocol, or RMCP
Note: In the ATCA-6002 shelf, RMCP is not enabled for the Base Ethernet interface. RMCP connections are enabled for the front panel (eth0), loopback (lo), and Fabric Ethernet (eth2) interfaces.
- IPMI over LAN through RMCP
- SNMP using the HPI-B0101-MIB module
- Platform-management command line interface through Telnet or SSH

Shelf Manager compliance

As of this manual's publication date, the Radisys Shelf Manager is fully compliant with the *PICMG 3.0 Revision 3.0 AdvancedTCA Base Specification*.

The Radisys hardware platform interface (HPI) is fully compliant with the mandatory requirements of the *SAF HPI-to-ATCA Specification*, SAI-HPI-B.03.02. The HPI is also fully compliant with the *SAF-ATCA Mapping Specification*, SAIM-HPI-B.01.01-ATCA.

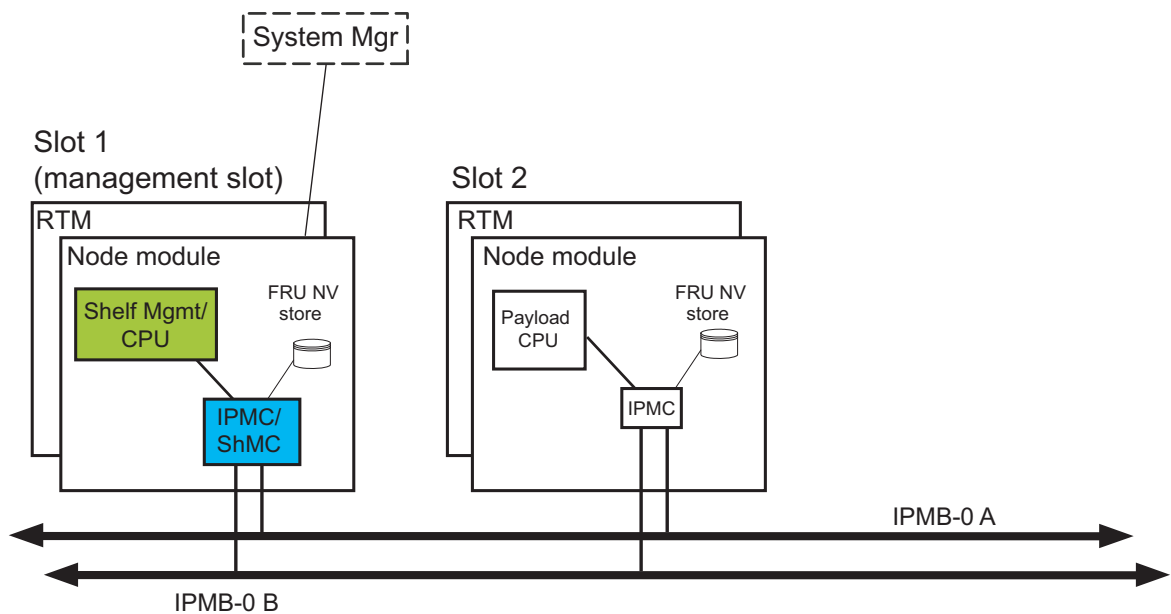
The Radisys Shelf Manager can manage any PICMG-compliant ATCA FRU. Radisys intelligent FRUs can also be managed by non-Radisys, PICMG-compliant Shelf Managers using IPMI.

For more details, see [Specifications and standards documents on page 8](#). For any updated compliance versions, see the Radisys ATCA Release Notes on the Radisys Web site.

Physical distribution of the shelf management infrastructure

[Figure 2](#) shows the physical distribution of the shelf management infrastructure on an ATCA-6002 shelf, where a supported module in slot 1 acts as the Shelf Manager. [Figure 3](#) shows the shelf management infrastructure on the remaining Radisys shelves, where an SCM acts as the Shelf Manager.

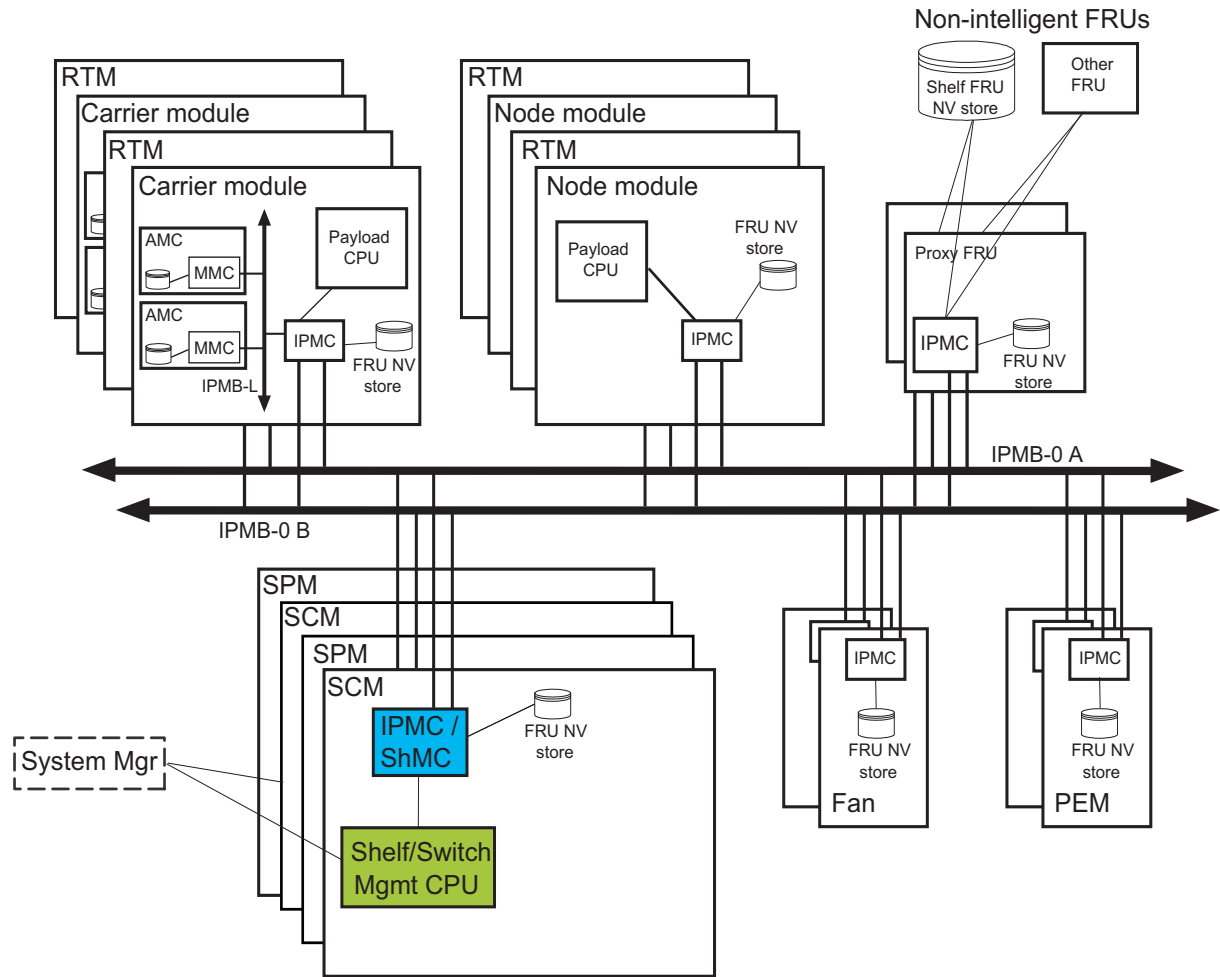
Figure 2. Shelf management components in ATCA-6002 shelves



IPMB — Intelligent Platform Management Bus
IPMC — Intelligent Platform Management Controller
Node module — Any front slot module other than the SCM

RTM — Rear transition module
ShMC — Shelf management controller
System Mgr — System management software

Figure 3. Shelf management component distribution in larger Radisys shelves



AMC — Advanced mezzanine card

Carrier module — Any front slot module that can host one or more AMCs

Fan — Cooling unit

IPMB — Intelligent Platform Management Bus

IPMC — Intelligent Platform Management Controller

MMC — Module management controller

Node module — Any front slot module other than the SCM

PEM — Power entry module

RTM — Rear transition module

SCM — Switch control module

ShMC — Shelf management controller

SPM — Shelf peripheral module

System Mgr — System management software

Major elements of the shelf management infrastructure

The following sections describe the major elements of the shelf management infrastructure.

Shelf Manager

The Shelf Manager watches over managed devices, reporting anomalous conditions to the system manager and taking whatever corrective actions it can to prevent system failure.¹ The Shelf Manager also manages hot-swap events, detecting the entry, removal, and shutdown of removable devices in the shelf.

The Shelf Manager can access data about the shelf capabilities and then manage the shelf and its FRUs (field replaceable units) so they operate within those parameters. For example, the Shelf Manager negotiates power budgets with FRUs to keep within the shelf's total power capacity. The Shelf Manager can also monitor events from FRU temperature sensors to detect when temperatures are outside of normal ranges and can then control fan operation in response.

The Shelf Manager functions are divided between the shelf management controller (ShMC) and the shelf management server (ShMS):

- The ShMC device contains firmware that performs most low-level shelf-management services. The ShMC device also performs IPMC functions on the Shelf Manager module.
- The ShMS software runs on the Shelf Manager module's local management processor and directly communicates with the ShMC. The ShMS performs the high-level shelf management functions.

IPMI and Intelligent Platform Management Bus (IPMB)

The IPMI architecture includes:

- Controllers that monitor and control resources in the shelf
- A bus for transporting messages (IPMB)
- A communications protocol
- A programming interface and message definition

The IPMB is based on inter-integrated circuit (I²C) bus technology, and is implemented as redundant buses in Radisys shelves. The IPMB provides the main communication link between the Shelf Manager and the intelligent devices in the shelf. The bus is called IPMB-0, with the redundant buses further identified as IPMB-A and IPMB-B.

A carrier module, like the ATCA-1200 Quad AMC carrier, has a local IPMB or IPMB-L for communicating with the advanced mezzanine cards (AMCs) it carries. The carrier's IPMB-L is separated from IPMB-0 both electrically and by use of a different address space.

1. Source: PICMG® 3.0 Revision 3.0 *AdvancedTCA Base Specification*, March 24, 2008.

Field replaceable units (FRUs)

AdvancedTCA defines two types of FRUs that are visible to and controlled through the IPMI infrastructure:

- Intelligent FRUs physically include an IPM Controller.
- Managed FRUs are either Intelligent FRUs or represented by an Intelligent FRU and visible to the IPMI infrastructure.

Each FRU that directly attaches to IPMB-0 must be an Intelligent FRU.² All Radisys ATCA front modules and AMCs are intelligent FRUs, along with most other FRUs in the shelf. The shelf itself is not a FRU. An intelligent FRU that represents a non-intelligent FRU is a proxy.

An example of a non-intelligent FRU is a shelf FRU device, which provides non-volatile storage for shelf-specific information. Access to the shelf FRU device through a proxy FRU is required for normal shelf management operation. Table 1 identifies the proxy FRUs and the non-intelligent FRUs they represent in each type of shelf.

Note: The ATCA-6002 shelf does not use proxy FRUs. This shelf does not support a physical shelf FRU device, SPM, PEM, or other devices that typically require proxy FRUs.

Table 1. Proxy FRUs and non-intelligent FRUs they represent in Radisys shelves

	ATCA-6000 shelf	ATCA-6006 shelf	ATCA-6014 shelf	ATCA-6016 shelf
Proxy FRU	SPM (one or two possible)	PEMs	RCM (Radisys Chassis Manager, one or two possible)	
Non-intelligent FRUs managed through proxy	Shelf FRU device Shelf display panel (SDP)	Two shelf FRU devices	Two shelf FRU devices (one per RCM) Shelf alarm panel Two PEMs Three fan trays	
Other entities managed through proxy	Shelf temperature sensors Air filter presence switches Telco alarm connector	n/a	Shelf temperature sensors Air filter presence sensor	
Additional information	When an ATCA-5010 SPM or ATCA-5014 SPM is installed, a virtual SPM provides generic management access by representing the currently active SPM at a fixed IPMB address. The virtual SPM address must be used instead of the active SPM address to communicate with the non-intelligent FRUs managed by the proxy FRU.	n/a	A virtual RCM provides generic management access by representing the currently active RCM at a fixed IPMB address. The virtual RCM reads data from only one of the two shelf FRU devices—the one corresponding to the active RCM. The virtual RCM writes updated data to both shelf FRU devices to keep their contents identical.	

For IPMB address information, see [IPMB addresses of slots and FRUs on page 134](#). For more SPM information, see the appropriate *Shelf Peripheral Module Hardware Reference* (ATCA-5010 or ATCA-5014). For more RCM information, see the *ATCA-6014/ATCA-6016 Platform Hardware Reference*.

2. Source: PICMG® 3.0 Revision 3.0 *AdvancedTCA Base Specification*, March 24, 2008.

Management controllers (IPMCs and MMCs)

Each intelligent FRU includes a management controller, either an Intelligent Platform Management Controller (IPMC) or a Module Management Controller (MMC). Most FRUs use IPMCs; however, AMCs and intelligent RTMs use MMCs instead. IPMCs and MMCs manage FRU resources such as sensors and controls.

IPMCs are the first shelf components to be powered up, allowing them to control power to the resources they manage. An IPMC can power off a functional block that is failing or that the Shelf Manager instructs to power off. The IPMC also gathers sensor data and communicates over the IPMB.

MMCs perform many of the same functions as IPMCs, including gathering sensor data and providing access to the AMC FRU inventory data. The MMC communicates with a carrier module's IPMC over the IPMB-L. After the carrier is powered and fully active, the payload of each AMC is powered up so it can control power to the rest of the AMC. The AMCs are powered up in order of site number, from lowest to highest. When a carrier module is extracted according to proper procedure, the AMC sites are shut down in the reverse order that they were activated (last site first). Then the carrier module is shut down and can be safely removed.

Sensor data records (SDRs)

Sensor data records (SDRs) control access to an IPMC's or MMC's managed FRU resources, including all sensors and FRU information. The Shelf Manager and system manager have knowledge only of sensors described by SDRs, and they gain access to them using Device SDR commands to query SDRs over the IPMB-0 interface.

Non-volatile information storage

Each intelligent FRU stores information about itself in non-volatile storage. The stored information includes:

- Serial number
- Part number
- Manufacturer
- Date and time of manufacture
- Product name
- EC/revision level (where EC is an abbreviation for engineering change)
- FRU capabilities
- Point-to-point connectivity records for modules that plug into the backplane (which are used for electronic keying)

Non-intelligent FRUs depend on a proxy FRU to store this information. In this case a proxy FRU makes the information available to the Shelf Manager.

The shelf contains a non-volatile storage device that stores shelf FRU information, including details of the shelf itself and the shelf configuration. The MultiRecord area of the shelf FRU information contains data such as shelf addressing, power management, and E-Key records. A proxy FRU makes this information available to the Shelf Manager. Redundant copies of the shelf FRU information are stored on the shelf management server (ShMS) file system.

Hardware platform interface (HPI)

Although IPMI over LAN can be used to communicate between the Shelf Manager and system manager, HPI provides a higher level interface that can be used generically as a management interface for highly available systems. Implementing the HPI interface in the Shelf Manager allows much of the autonomous shelf management functionality to be terminated at the Shelf Manager and a clean, portable interface to be presented to the system manager using HPI. Furthermore, this HPI interface allows system management software to be easily portable to any HPI-compliant platform, including non-AdvancedTCA platforms such as CompactPCI.

For a description of the HPI architecture, see [HPI architecture on page 20](#). For information on using HPI, see [Chapter 4, Changing Shelf Settings Using HPI, on page 53](#).

System manager

A system manager is a high-level management application that may manage entities ranging from a single shelf to multiple systems. The system manager uses the capabilities and interfaces provided by shelf management to manage the ATCA shelf. Although a system manager is not delivered with Radisys ATCA products, the interface between the Shelf Manager and a system manager is described in this manual.

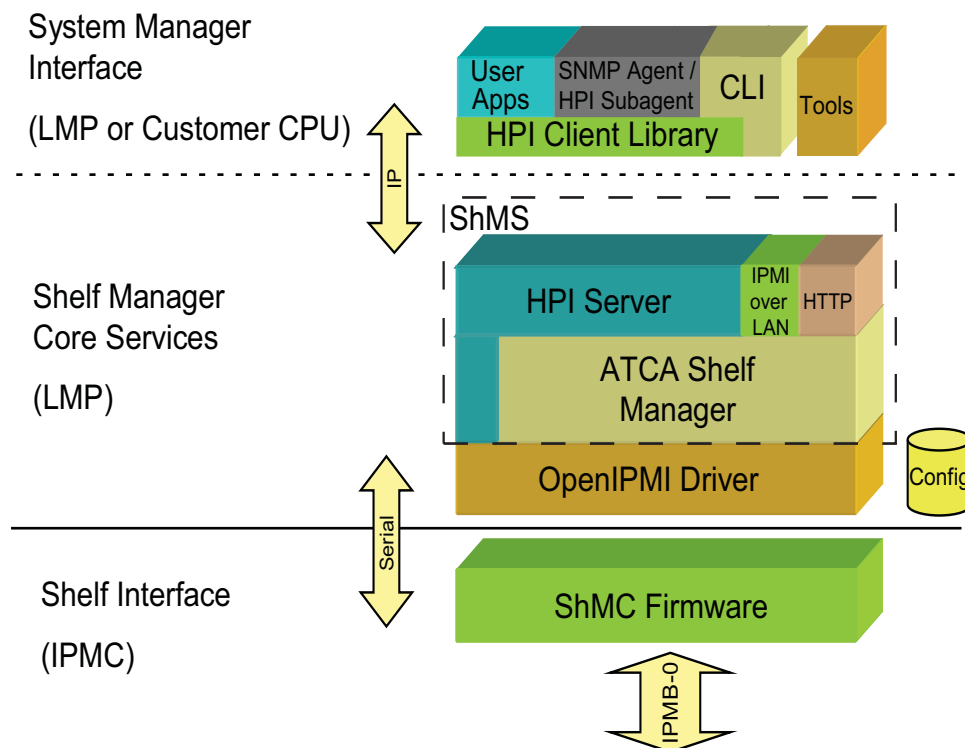
Software Architecture

Shelf management architecture

Shelf management services for the Radisys platform are partitioned across two processors: the local management processor (LMP) and the Intelligent Platform Management Controller (IPMC).

- High-level functions are contained within the shelf management server (ShMS) and execute on the LMP.
- Low-level functions are contained within the shelf management controller (ShMC) and execute on the IPMC.
- An application programming interface (API) allows management applications to remotely access shelf management features available on the Shelf Manager module. The following figure depicts the overall software architecture of Radisys shelf management.

Figure 4. Shelf management software architecture



Shelf interface

At the shelf interface level, the ShMC firmware runs on the IPMC, performing the low-level shelf management operations as well as the IPMC responsibilities for the Shelf Manager module FRU. The shelf interface services include:

- System event logging (SEL)
- Sensor data records
- Sensor monitoring
- IPMB monitoring
- FRU inventory
- E-Keying
- Power and hot-swap management
- FRU presence detection (heartbeat)

Shelf Manager core services

The ShMS, a daemon that runs on the LMP, provides interfaces for the Shelf Manager and the system manager. The HPI server is integrated with the ShMS, which hosts the HPI system management interface. HPI applications can access HPI services both locally on the Shelf Manager module and remotely over IP. The ShMS supplies an IPMI-over-LAN interface that allows IPMI applications and CLI users to directly access IPMI and shelf management services.

The ShMS uses locally stored, persistent configuration files to set default and user-customized configuration values. It also provides diagnostic utilities for troubleshooting purposes.

System manager interface

The HPI client library (HCL) and the IPMI-over-LAN interface allow user applications, including a system manager, to manage the Radisys shelf remotely. You can use the OpenHPI library and the applications it provides to assist in developing HPI applications for the Radisys shelf. Radisys also provides a Simple Network Management Protocol (SNMP) access to HPI services and an HPI example application called *hpiapp*.

In addition, the command line interface (CLI) allows interactive monitoring, debugging, and intervention. Utilities enable remote upgrades of the Shelf Manager software and firmware, as well as updates to the shelf FRU device data.

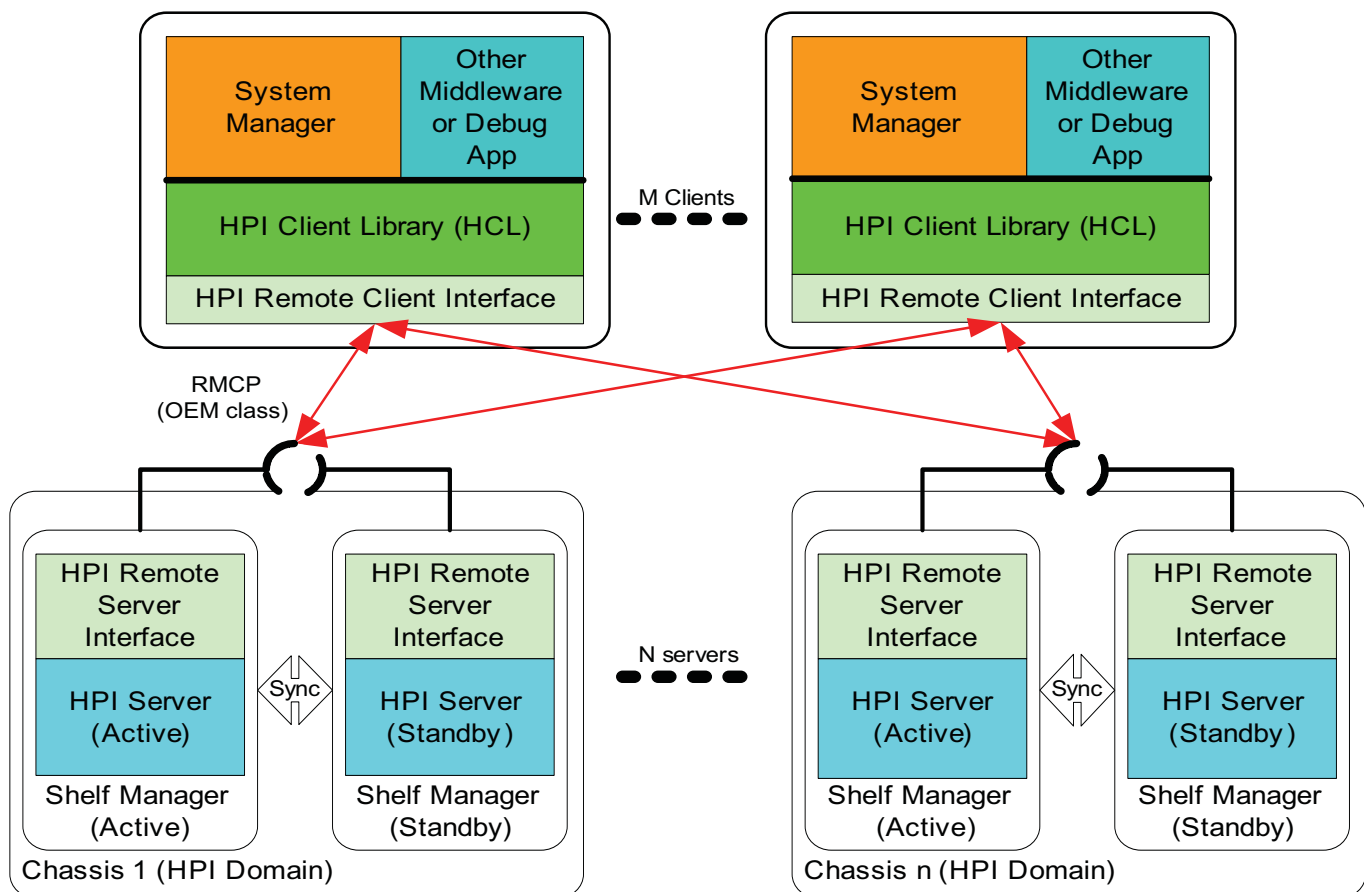
HPI architecture

The primary management interface to the Radisys shelf management services is the Hardware Platform Interface (HPI). The Intelligent Platform Management Interface (IPMI) is the lower-level infrastructure and interface that the HPI relies upon to carry out tasks.

The Shelf Manager supports integrated HPI client-server services that are compliant with the *SAF HPI Specification*. This section assumes familiarity with the HPI model section of the specification.

In addition to shelf management, HPI also provides a common interface for blade management. The HPI framework enables payload resources on each module's controller to be available and managed through the Shelf Manager, providing a single point of access to all of the resources in the shelf. For example, HPI Firmware Upgrade Management Instruments (FUMIs) are implemented on the modules to support Radisys software upgrades. Through HPI, software upgrades on all modules can be managed remotely on the Shelf Manager. Radisys implements this using a blade HPI server (rsyshsd) on each supported ATCA front module. (For more information on HPI based Radisys software upgrade structure, see the *Software Guide*.)

Figure 5. HPI service architecture

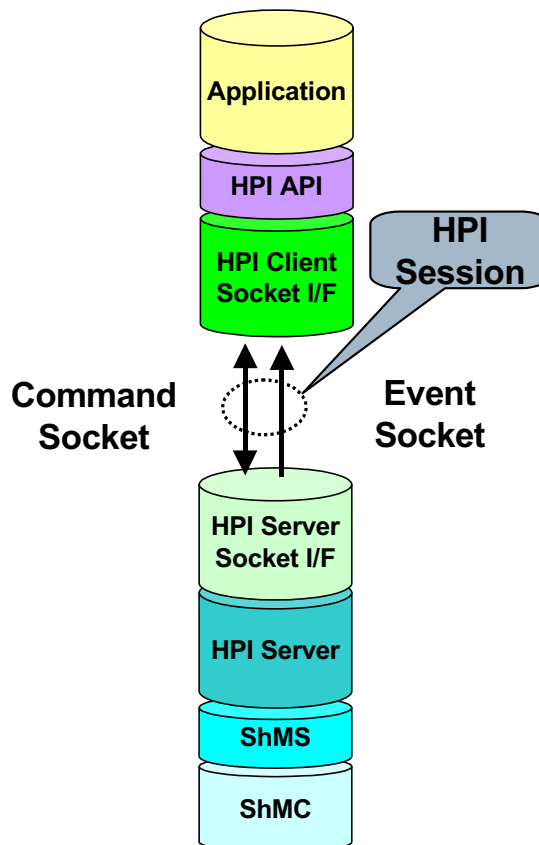


As shown in [Figure 5](#), the HPI client library (HCL) is a dynamically linked library (DLL) that provides the HPI API as well as an integrated RMCP client for remote HPI client-server communication. The HCL source code is independent of the operating system and management processor, and works with any generic HPI-compliant application. The HCL Linux library is installed on the Shelf Manager module. The HCL source code is available in the Radisys software distribution.

An SNMP agent is integrated with the HCL in order to provide HPI services to an SNMP management application. To enable the HPI subagent, see [Enabling the HPI subagent on page 61](#). For the specific SNMP objects to use, see [Chapter 4, Changing Shelf Settings Using HPI, on page 53](#).

The HPI client library communicates with the HPI server through a remote socket interface (see [Figure 6](#)). The HCL encapsulates HPI request messages into RMCP packets, which are delivered over User Datagram Protocol (UDP). The HPI server is resident on the Shelf Manager and accepts remote commands received at the RMCP server port, 623.

Figure 6. HPI implementation



RMCP, a request-response protocol, is a simple packet-based communication mechanism. RMCP packets include a field that indicates the class of message embedded in the packet. The messages between the HPI client and HPI server are encapsulated in RMCP packets using the OEM class. For more information, see the RMCP section of the *Intelligent Platform Management Interface Specification v2.0, Document Revision 1.0*.

The HPI service reports the system status from the perspective of the system it is running on. When a shelf has redundant SCMs, the HPI server also employs an active/standby redundancy model. In this model, the active HPI server runs on the active Shelf Manager and the standby HPI server runs on the standby Shelf Manager. For more information, see [HPI server redundancy on page 31](#).

In a run-time environment, the HPI client library is considered an extension of the management application. The remote network interface provided by the client library allows the management application to execute from anywhere within the network and on any module running the Linux operating system. The client library may execute on the same module as the Shelf Manager. In a local environment, the client still uses the socket interface but in its local form.

Domain alarm table (DAT)

HPI maintains a domain alarm table of active alarms related to the shelf. The table's current alarms cause the alarm LEDs to turn on and off, unless the system manager overrides this behavior. Unacknowledged alarms additionally cause the audible alarm to sound, unless the system manager overrides this. Alarms are removed from the table when the condition is resolved.

For information on interacting with the DAT, see [Responding to alarms on page 100](#). For details on how the Shelf Manager responds to alarms, see [Managing Alarms and Events on page 82](#). For general information about the DAT, see the domain alarm table section of the *SAF HPI Specification*.

System event log (SEL)

The IPMC of each intelligent FRU keeps a local system event log containing all events generated by the IPMC. Some events announce state changes, and others indicate health conditions and cause alarms. The SEL on the active Shelf Manager module acts as the SEL for the entire shelf. It is a collection of the SEL entries from all FRUs in the shelf, and identifies each event by the IPMB address of the FRU that generated it (in the least significant byte of the generator ID).

The SEL is primarily intended for troubleshooting purposes and does not offer a programmatic interface. The similar [Domain event log \(DEL\)](#) can be accessed programmatically by a system manager. The SEL contains a complete list of IPMC events, so it is the best source for events such as system progress checkpoint events generated by Radisys modules, which are excluded from the domain event log.

For information on viewing the log messages, see [Viewing the system event log on page 101](#). Events may be cleared from the shelf's SEL when the Shelf Manager is restarted, depending on the configuration explained in [Configuring the system event log \(SEL\) on page 45](#)

Domain event log (DEL)

HPI provides a domain event log that contains most of the events that are in the shelf's SEL. The domain event log includes only events that are associated with a sensor for which the IPMC presents an SDR. This excludes all events associated with virtual sensor 0xff, such as the system progress checkpoint events that are generated as the Radisys front modules boot up. These events can be found in the SEL.

The domain event log is easy to read and access. It can be accessed by a system manager programmatically, and can also be helpful for troubleshooting purposes when accessed interactively. Each event is identified by the resource ID of the module that generated it. Users can add events to the domain event log. Events are kept in the domain event log (even if the condition is resolved) until the log is erased or the Shelf Manager module is restarted.

For information on viewing the log messages, see [Viewing the domain event log on page 101](#). To configure the behavior of the domain event log when it reaches its capacity, see [Configuring HPI domain event log settings on page 48](#). For general information about the domain event log, see the event log management section of the *SAF HPI Specification*.

Basic and enhanced shelf management operation

The Shelf Manager in an ATCA-6002 shelf can operate in two modes: enhanced and basic. By default, the Shelf Manager operates in basic mode. For details on operating in enhanced mode, refer to [Changing the Shelf Manager operating mode on page 39](#).

Note: The ATCA-45xx series and the ATCA-46xx series CPMs only operate in the enhanced mode when they are installed in the ATCA-6002 shelf.

Basic mode

In basic mode, the Shelf Manager provides limited shelf management functionality in the ATCA-6002 shelf. The Shelf Manager does not support hot swap or manage any events and alarms in basic mode.

The Shelf Manager provides payload power to the module on which it is running in slot 1, and then enables power to the node module in slot 2. The Shelf Manager activates subsidiary FRUs on both modules and the backplane interfaces on the node module. It then hands over control of the shelf to the system management software. The IPMC firmware continues to monitor the state of the modules.

The Shelf Manager uses a configuration file (`/etc/shmgr_basic.conf`) that contains static information about the possible node modules that can be installed in slot 2. This file includes the E-Key parameters and key sensor numbers for scripts used to activate the subsidiary FRUs and backplane E-Keys. For details, refer to [Configuring the Shelf Manager in basic mode on page 50](#).

Enhanced mode

In enhanced mode, the Shelf Manager provides full shelf management functionality in the ATCA-6002 shelf, with these exceptions:

- No redundant Shelf Managers.
- No proxy FRUs.
- No controllable fans in the shelf. The Shelf Manager disables the cooling algorithm.
- No physical shelf FRU device. A persistent shelf FRU file (`/var/lib/shmgr/atca-6002.bin`) is stored on the Shelf Manager module.
- No physical alarm support (audible or visual). Alarms are still added to the HPI domain alarm table.
- No HPI SNMP subagent support.
- No RMCP connections through the Base Ethernet interface. RMCP connections are enabled for the front panel (eth0), loopback (lo), and Fabric Ethernet (eth2) interfaces.

The Shelf Manager enables power to the module on which it is running, and enables payload power and E-Keys for the node module in slot 2. The user's applications interact with the Shelf Manager through the HPI API.

In enhanced mode, the Shelf Manager uses the */etc/shmgr.conf* file. The Shelf Manager does not need to use the *shmgr_basic.conf* file provided for basic mode; it can discover and manage any type of ATCA node module. The *rsys-ipmitool* utility is also available for performing operations at the command shell and in scripts.

Software redundancy

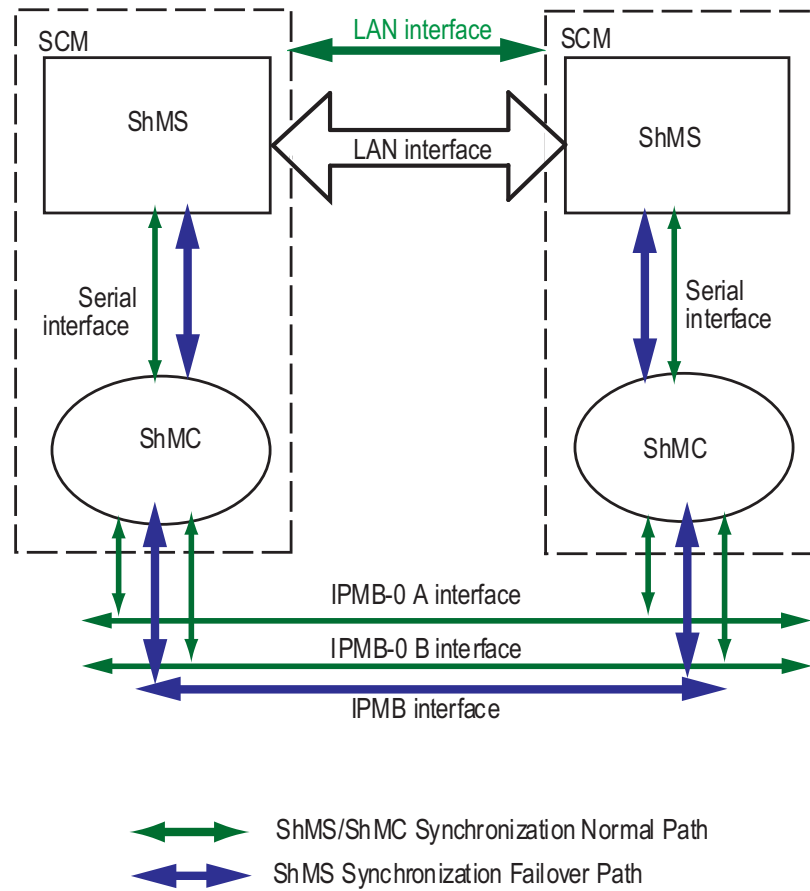
The following sections apply only to shelves where an SCM is acting as the Shelf Manager. They do not apply to ATCA-6002 shelves.

Shelf Manager redundancy

The Shelf Manager operates in a fully redundant active-standby model when both SCMs are installed. During system startup, if redundant Shelf Managers are detected, an arbitration mechanism between the peer ShMCs decides which one takes over as active while the other assumes the standby role. Once arbitration is complete and the active Shelf Manager is initialized, the Active LED on the active Shelf Manager glows solid amber. The active Shelf Manager is responsible for monitoring and managing the shelf. All configuration changes and event data are synchronized with the standby Shelf Manager so that it is ready to assume the active role at any point of time.

On a single SCM, the ShMS running on the LMP and the ShMC communicate with each other over a serial link using the OpenIPMI serial driver in Basic mode. The two ShMCs from the SCM pair communicate over the IPMB, which is physically redundant. Additionally, the two ShMSs from the SCM pair synchronize over an Ethernet LAN interface on the backplane.

Figure 7. SCM Shelf Manager redundancy



Shelf Manager failover

Component failures in the ShMS and ShMC are detected through the following combination of sending an echo request packet then waiting for a valid response (pinging) and monitoring the periodic watchdog strobes:

1. Each ShMC pings its peer ShMC over the IPMB.
2. Each ShMS attempts to ping its peer ShMS over the LAN interface on the backplane.
3. If pings over the LAN interface fail, the ShMS tries to ping its peer ShMS using the failover path through IPMB and the serial interfaces, as indicated in [Figure 7](#).

Peer communication loss causes failover

If communication between the peer ShMCs breaks down, the first of the previous communication mechanisms (ShMC-to-ShMC over IPMB) fails. The standby Shelf Manager attempts to ping its peer over the LAN interface to confirm its full high availability (HA) state. If this fails, this causes the standby Shelf Manager to effect a failover, assume the active role, and send out an event notifying all event receivers of the failover. This can happen if the active Shelf Manager was hot-swapped from the shelf improperly or if the local IPMB on the active Shelf Manager hardware failed, causing it to de-link itself from IPMB-0.

Watchdog expiration causes failover

If the operating environment that the ShMS is running on fails, the IPMI watchdog in the ShMC expires, causing it to reset the module hosting the ShMS so that it reboots to a good state. If this happens in the active Shelf Manager, a failover to the standby Shelf Manager occurs and the previously active Shelf Manager assumes the standby role after it reboots. However, if this happens in the standby Shelf Manager, then no failover is required, but a payload reset to a good working standby mode still happens.

Peer communication loss causes failover, non-redundancy

If the peer ShMSs are not able to maintain communication over the LAN interface, they attempt to synchronize over the IPMB. If that fails as well, the standby ShMS assumes the active role. This protects against an unlikely condition where the active ShMS has an unrecoverable failure, but keeps running in a non-functional state. With communication over the LAN interface lost, the Shelf Managers are no longer considered redundant, and the [Shelf Manager redundancy sensor on page 28](#) issues an alarm.

Watchdog expiration causes reboot of non-redundant SCM

If no standby Shelf Manager is present and the communication between the active ShMS and ShMC breaks down, the ShMC watchdog still expires, causing a software reboot of the SCM. In this case, the Shelf Manager still reboots to a good working state and reassumes the active role. The reboot results in downtime, because there is no ShMS to monitor and respond to events from shelf components while the Shelf Manager payload is rebooting.

Initiating a failover manually

You can manually initiate a failover by:

- Using the platform-management CLI.
- Using HPI - Control number 0x1010 in resource 0x02. Refer to the Shelf Manager failover control section of the *SAF Mapping Specification* for more information.
- Toggling (opening and then closing) the bottom ejector latch of the active SCM.
- Powering off or power cycling the active SCM using HPI with the *saHpiResourcePowerStateSet()* function.
- Extracting the active SCM either manually or using HPI with the *saHpiHotSwapActionRequest()* function.

- Using the front panel reset button to reset the active SCM either manually or using HPI with the `saHPIResourceResetStateSet()` function.

When the SCM is acting as the active Shelf Manager, its Active LED glows amber. On the standby Shelf Manager, this LED flashes periodically, about once every 2 seconds. During a switchover operation in which the roles of the Shelf Managers are swapped, the LED flashes rapidly on the Shelf Manager that is moving from standby mode to active mode until the switchover is complete. A rapidly flashing amber LED indicates that the Shelf Manager is preparing its high availability role. After a shelf power up or an SCM reboot, you might see the amber LED flashing rapidly before it becomes steady in active mode or slowly blinking in standby mode.

Shelf Manager redundancy sensor

The HPI Shelf Manager resource contains a standard sensor that monitors the Shelf Manager's redundancy status. The Shelf Manager redundancy sensor is a discrete sensor that indicates if the Shelf Manager is redundant at any point of time. For descriptions, see the virtual Shelf Manager resource RPT entry section of the *SAF Mapping Specification*.

The Shelf Managers are considered redundant when the following conditions are met:

1. The active Shelf Manager has established the presence of the standby Shelf Manager using a dedicated IPMI over a LAN session over the synchronization channel.
2. The standby HPI server has successfully completed the initial *refresh sync* with the active HPI server.
3. The active and standby Shelf Managers have established that they are running the same software version.

When all of the above conditions are met, the Shelf Manager redundancy sensor reports the event state `SAHPI_ES_FULLY_REDUNDANT`, signifying full redundancy has been achieved.

If the first two conditions are not met, the Shelf Manager redundancy sensor reports the event state `SAHPI_ES_REDUNDANCY_LOST_SUFFICIENT_RESOURCES`, signifying that the Shelf Managers are currently not redundant, but the active Shelf Manager is still capable of managing the shelf. Associated alarms are created when redundancy is lost between the Shelf Managers.

If the third condition is not met, the Shelf Manager redundancy sensor reports the event state `SAHPI_ES_REDUNDANCY_DEGRADED`, signifying that the Shelf Manager is still fully functional, but that any new features available with the later software version are disabled. This condition is not considered an alarm and no alarm is logged; however, an event is generated.

Shelf FRU information redundancy

During normal shelf operation with redundant Shelf Managers, there are always at least three persistent copies of the shelf FRU information. The shelf FRU information is initially stored in a single shelf FRU device in an ATCA-6000 shelf, and in two shelf FRU devices in an ATCA-6006, ATCA-6014, or ATCA-6016 shelf. In addition, both the active and standby Shelf Managers keep a persistent cache of the shelf FRU information. One of the first steps taken by the active Shelf Manager during initialization is locating, validating, and creating or updating these persistent caches of the shelf FRU information.

The active Shelf Manager validates the checksum of each shelf FRU information file and also compares the contents of the files to resolve any conflicts. If a conflict exists, the Shelf Manager gives priority to the shelf FRU device contents over the copies cached by the Shelf Manager. Once the Shelf Manager identifies good content, it copies over the content of the other locations if their contents differ.

The Shelf Manager uses these basic rules to select good content:

- If a shelf FRU device's checksum is valid (and its content does not conflict with another shelf FRU device on the ATCA-6006 shelf), the Shelf Manager selects the shelf FRU device content.
- If no shelf FRU device has a valid checksum but the copies cached by both the active and standby Shelf Manager have valid checksums and matching content, the Shelf Manager selects the active Shelf Manager's cached content.
- If two shelf FRU devices have conflicting content, the Shelf Manager compares their content to the Shelf Manager-cached copies and selects the shelf FRU device content that matches more cached copies.

Note: This rule applies to the ATCA-6006 shelf only. Although the ATCA-6014 and ATCA-6016 shelves each have two shelf FRU devices, only one is read—the device accessible to the active RCM. The contents of the two devices are not compared.

- If valid shelf FRU information cannot be located or a conflict cannot be resolved, the Shelf Manager generates a shelf FRU Invalid alarm to request operator assistance in updating the shelf FRU information.

Shelf FRU invalid alarm

This alarm signals a catastrophic fault condition. The active Shelf Manager keeps all other modules in the M2 (pre-activation) state and waits for the operator to intervene.

On the ATCA-6000 shelf, the “Shelf FRU Invalid” text is visible on the shelf display panel. On any Radisys shelf, both SCMs are placed into debug mode. This banner is shown upon login:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                                                                    !!
!! WARNING: There was a problem upon bootup. This blade !!
!!           is now in DEBUG mode so the error(s) can be !!
!!           corrected.                                     !!
!!                                                                    !!
!!           THIS BLADE IS NOT CONSIDERED TO BE IN A      !!
!!           USABLE STATE WHEN IN DEBUG MODE.             !!
!!                                                                    !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
*** Boot error logged: ***
```

```
/sbin/fruinfo failed to retrieve valid information from shelf manager
```

This alarm can only occur immediately after SCM insertion. To correct the situation, see the troubleshooting topic [Shelf Manager module boots up in debug mode on page 126](#).

For other reasons for entering debug mode, see the *Software Guide*.

HPI server redundancy

The Radisys HPI server uses a redundancy model very similar to the active-standby model used by the Shelf Manager. Since the HPI server is combined with the Shelf Manager Server (ShMS) daemon, it shares the same redundancy status with the Shelf Manager. The active Shelf Manager hosts the active HPI server while the standby Shelf Manager hosts the standby HPI server. At any point of time after HPI server initialization is complete, the standby HPI server is fully redundant with the active HPI server and is capable of assuming the active role immediately following a failover.

The active HPI server fully complies with the *SAF HPI Specification*. The fully redundant standby HPI server cannot perform certain actions, such as hot-swap handling and failovers due to platform and implementation restrictions placed on the standby role. Although it is possible to open sessions on the standby HPI server for normal operation, HPI sessions should be opened only on the active HPI server using the dedicated Shelf Manager IP address as DomainId. See [Writing HPI applications on page 57](#) for information on how to open HPI sessions.

The following types of HPI data are actively synchronized with the peer HPI servers.

- Sessions – Sessions opened on the active HPI server persist over failovers to the standby Shelf Manager and HPI server. HPI applications can use the same SessionId to execute HPI operations on the system following a failover. HPI applications that opened the session using the dedicated Shelf Manager IP address as DomainId will not experience any noticeable difference in HPI server operation following a failover, other than possibly a small delay in the next HPI command issued following the failover.
- Event subscriptions – Session event subscriptions persist over failovers. If an event subscription was opened on a session, it remains in effect following a failover. Session event queues are also synchronized, ensuring that no events are lost from a session event queue that has subscription enabled due to a failover.
- Domain tag, resource tag, and resource severity – All user changes to the domain tag, resource tags, or resource severities are synchronized to the peer HPI server and persist following a failover.
- Events and event logs – All events, both system generated and user added, are synchronized to the standby HPI server. The HPI domain and resource event logs in the active and standby HPI servers are thus kept in sync. User changes to event log times and event log states are also synchronized.
- Alarms – All alarms, both system generated and user added, are synchronized to the peer HPI server. The domain alarm tables in the active and standby HPI servers are thus kept in sync. User initiated changes, such as alarm acknowledgements and user alarm deletions, are also synchronized.

In addition to the HPI data, automatic operations are synchronized with the peer HPI servers to keep HPI databases in sync. These operations are called by the following functions and are only synchronized with the standby HPI server when the HPI application makes the request to the active HPI server.

- *saHpiDiscover()* – This function overcomes potential latencies in the hot-swap mechanism and keeps the resource presence table (RPT) updated. This operation is synchronized to the peer HPI server so it can update its RPT.

Note: The *saHpiDiscover()* function is not meant to perform any initialization or discovery actions and should not be used for an HPI implementation. However, the function can be used in debug situations (refer to [Table 15 on page 126](#) for examples).

- *saHpiEventLogClear()* – This function clears the specified HPI event log in both active and standby HPI servers.
- *saHpiParmControl()* – This function saves custom configurations for resources. The configurations are synchronized with the peer HPI server to ensure local HPI configuration files are updated.

During HPI server initialization, the standby HPI server sends a *refresh* request to the active HPI server over the synchronization medium. Upon reception of this request, the active HPI server sends any data that needs to be synchronized to the standby HPI server. The standby HPI server becomes fully initialized and the HPI service becomes fully redundant once the *refresh sync* with the active HPI server is completed.

The dynamic redundancy between the active and standby HPI servers enable the system manager to get a consistent view of the Radisys system regardless of failovers, and ensures that a system management interface is always available.

Shelf cooling

To ensure that all modules in the shelf operate efficiently, their operational temperatures must be maintained at optimal levels. All modules affected by on-board temperature fluctuations have sensors that generate events in case their operational temperatures cross pre-set threshold levels. Temperature sensors generate events at three different threshold levels: Minor, Major, and Critical.

The active Shelf Manager registers as an event receiver with all intelligent shelf components in order to receive all temperature events generated by sensors in the shelf. It uses a cooling algorithm to make adjustments to the cooling hardware to maintain optimal component temperatures while minimizing noise levels.

Summary of supported cooling hardware

[Table 2](#) identifies the cooling hardware for each Radisys shelf. In this document, all types of cooling hardware are referred to generically as a fans.

Table 2. Cooling hardware controlled by the Shelf Manager

Radisys shelf	Cooling hardware
ATCA-6000	Four air mover modules (AMMs), each controlled separately.
ATCA-6006	Two fan trays, each controlled separately. Each fan tray contains multiple fans that cannot be separately controlled.
ATCA-6002	The fans in this shelf always run at full speed and are not under Shelf Manager control.
ATCA-6014	Three fan trays that contain multiple fans. All of the fan trays in the shelf are controlled as a single unit.
ATCA-6016	

Shelf Manager cooling algorithm

At shelf power-up, the fans run at close to full speed as a safety precaution against overheating before the Shelf Manager takes control. The active Shelf Manager then programs all fans to operate at a percentage of their maximum speed before it starts receiving temperature events. Once the active Shelf Manager is fully initialized, the fan speeds are reduced and remain there until a temperature event is received. A fan floor setting determines the lowest fan speed the Shelf Manager will ever set.

Minor and major temperature event assertions cause the fan speeds to increase in increments until the condition is cleared. The Shelf Manager raises the fan speed one level for a minor temperature event and two levels for a major event. A de-assertion event from the same sensor indicates that the condition is cleared, and the fan speed is decremented one level at a time—as long as no temperature events are asserted—back to the level of the fan floor. Fan speed changes are reduced over time by adjustments to the fan floor, as explained in [Adaptive fan floor algorithm on page 34](#).

When the Shelf Manager receives a critical temperature event, it attempts to power off the module associated with the sensor. Even while powered off, the sensors are still functional. The alarm condition may not be cleared immediately, if at all, so the fans may continue to run at a high speed.

If both SCMs are removed from the ATCA-6014 or ATCA-6016 shelf, the active RCM adjusts the fans to full speed until the Shelf Manager resumes managing the shelf.

When a fan module is deactivated (by toggling its hot-swap switch or performing the software hot-swap procedure through the CLI), the motor will be turned off. A fan whose switch is in the wrong position at shelf power up will immediately be powered off as soon as the Shelf Manager initializes.

Adaptive fan floor algorithm

An adaptive fan floor algorithm attempts to determine a fan floor level that minimizes both fan noise and changes to fan speeds while still achieving the needed cooling. The Shelf Manager will not lower the fan speed below this floor.

The Shelf Manager has a default setting of 30 percent for the fan floor, and users can manually set a higher fan floor as described in [Setting the minimum fan speed on page 45](#). The fan floor algorithm may determine a new fan floor value above the default or user-specified floor, which could be up to the maximum floor of 80 percent.

The fan floor algorithm runs every 30 minutes and takes into account any temperature sensors that have generated threshold-crossing events as well as the number of fan level increases and decreases during that time. The algorithm is likely to increase the level of the fan floor if the fan speed has increased and decreased several times during the 30-minute window. It will decrease the fan floor to reduce the noise if no temperature alarms were generated and if the floor is not already set to the minimum.

The current fan floor setting can be read from the Shelf Fan Floor Level sensor, which is described in [Table 20 on page 146](#). In addition, the results of each run of the fan floor algorithm and the corresponding fan floor setting are shown in the Shelf Manager log files (*/tmp/shmgr/shmgr.log*.tgz*). To find the fan floor information, search the log file for this string:

```
SHMS<Cooling>
```

The sensor and log file report the fan floor level as a percentage of the maximum fan speed.

Shelf Manager cooling override

The Shelf Manager provides autonomous cooling management for the shelf. However, there may be situations under which a system manager might need to override the Shelf Manager's cooling management and control fan speeds itself. HPI applications can change fan speeds incrementally using standard HPI controls in the fan resources. For more details on the HPI controls, see the cooling and fan control section of the *SAF Mapping Specification*.

Each fan resource has one fan control in its resource data record (RDR) repository. In the ATCA-6000 and ATCA-6006 shelves, system managers may choose to use the individual fan controls to operate only certain fans manually while letting the Shelf Manager operate the others. This can be useful when the system manager wishes to override the Shelf Manager cooling management only on one side of the shelf that hosts specific hardware.

Note: In the ATCA-6014 and ATCA-6016 shelves, all fans are controlled together. Making a change to any of the fan controls affects all of the fans.

System managers can also choose to completely override the cooling algorithm and use an external cooling module to control cooling. This is described in [Overriding the cooling algorithm on page 74](#). Overriding the cooling algorithm also prevents any fan speed changes caused by the adaptive fan floor algorithm; however, the algorithm could set a new fan floor, which would take effect if cooling control were returned to the Shelf Manager.

Configuring Shelf Manager and HPI Behavior

This chapter explains how to:

- Disable or enable the Radisys Shelf Manager.
- Change the operating mode of the Shelf Manager in the ATCA-6002 shelf.
- Configure Shelf Manager and HPI behavior by editing the `/etc/shmgr.conf` file.
- Configure Shelf Manager behavior in basic mode in the ATCA-6002 shelf.
- Configure blade HPI behavior.

Disabling or enabling the Shelf Manager

By default, the Shelf Manager is enabled, which is appropriate when the module on which it is running is installed in a Radisys ATCA-6xxx shelf. The following table lists scenarios for when you should disable or enable a module's Shelf Manager.

Table 3. Scenarios for disabling or enabling the Shelf Manager

Module	When to disable the Shelf Manager	When to enable the Shelf Manager
SCM	<ul style="list-style-type: none"> • To install the SCM in a non-Radisys shelf • To install the SCM in the Radisys MPCH0001 shelf 	To install the SCM in a Radisys shelf after removing it from a non-Radisys shelf or Radisys MPCH0001 shelf ¹
Shelf Manager module in an ATCA-6002 shelf (For the list of supported modules, refer to Shelves and modules supported on page 7 .)	<ul style="list-style-type: none"> • To install the module in a non-Radisys shelf • To install the module in any Radisys shelf other than slot 1 of the ATCA-6002 shelf 	To install the module in slot 1 of an ATCA-6002 shelf after removing it from a non-Radisys shelf or any other Radisys shelf ¹
¹ Before removing the module, you must enable the module's Shelf Manager.		

The following procedures describe how to disable the Shelf Manager and enable it if it was previously disabled.

Important: You need *rsys-ipmitool* version 1.8.9, build 2.41 or greater to perform the Shelf Manager configurations. Install the latest *rsys-ipmitool* from the `rsys-ipmitool-1.8.9-241<version>.rpm` (or greater) provided with your Radisys software release.

Disabling the Shelf Manager

Important: When there is only one SCM or Shelf Manager module in the shelf, disabling its Shelf Manager causes all shelf management functionality to be lost. Additionally, if you power cycle the shelf or the module with the disabled Shelf Manager, all modules in the shelf (including the SCM or Shelf Manager module) will be unable to power up their payloads. To re-enable the Shelf Manager and allow the payload to power up, you must install the SCM or Shelf Manager module into a shelf with an active Shelf Manager running and then re-enable it.

1. Establish a serial connection to the Shelf Manager module and log on as root. See the module's *Installation Guide* for details.

2. At the prompt of the connected computer, the following may appear:

```
DEBUGMODE :~#
```

This DEBUGMODE prompt is a result of a conflict between the Shelf Manager on the module and the Shelf Manager on the non-Radisys shelf. Disabling the Shelf Manager on the module resolves this conflict.

3. Disable the Shelf Manager by entering:

```
rsys-ipmitool rsysoem shmc disable
```

4. To verify that the module's Shelf Manager is disabled, enter:

```
rsys-ipmitool rsysoem shmc info
```

If the Shelf Manager is disabled, the response is ShMC state: Disabled

5. If a second SCM is installed, perform steps 1 through 4 on the second SCM as well.
6. Cycle the power on the shelf to restart the non-Radisys Shelf Manager.
7. See the module's *Installation Guide* to complete the verification steps.

Enabling the Shelf Manager

Important: If you are removing a module from a non-Radisys or Radisys MPCH0001 shelf and reinstalling it in an ATCA-6xxx shelf, you must first enable the module's Shelf Manager before removing it. Otherwise, you will need to install the module in a shelf (Radisys or non-Radisys) that already has an active Shelf Manager running.

1. Establish a serial connection to the module and log on as root. See the *Installation Guide* for details.
2. Enable the Shelf Manager by entering:

```
rsys-ipmitool rsysoem shmc enable <address1> <address2>
```

For example, to enable an SCM at the two hub slot addresses:

```
rsys-ipmitool rsysoem shmc enable 0x82 0x84
```
3. To verify that the module's Shelf Manager is enabled, enter:

```
rsys-ipmitool rsysoem shmc info
```

If the Shelf Manager is enabled, the response is ShMC state: Active
4. If applicable, remove the module and reinstall it in a Radisys shelf. For removal instructions, see the module's *Reference*. For installation instructions, see the module's *Installation Guide*.
5. If you are planning to remove a second SCM from the non-Radisys shelf, connect to it and run through the same set of procedures described in steps 1 through 4.

Configuring multiple modules

Multiple modules installed in a shelf can be configured at the same time by generating a script that executes the *rsys-ipmitool* command several times. The script could be run from a head system through an RMCP connection.

For example, this is the command syntax to disable the Shelf Manager:

```
rsys-ipmitool -H <Shelf Manager/RMCP server IP address> -A <authentication type>  
-t <IPMB address of module to configure> rsysoem shmc disable
```

Specifically, the following command disables the ShMC for a CPM at address 0x86 through an ATCA-2210 running Shelf Manager at IP 10.100.19.242:

```
rsys-ipmitool -H 10.100.19.242 -A none -t 0x86 rsysoem shmc disable
```

To script the process, multiple invocations of the *rsys-ipmitool* command could be called in the script, with each invocation referencing a different module IPMB address.

Changing the Shelf Manager operating mode

The Shelf Manager in an ATCA-6002 shelf can operate in two modes: enhanced and basic. For a description of these operating modes, refer to [Basic and enhanced shelf management operation on page 24](#). By default, the Shelf Manager runs in basic mode for most of the modules. The exception to this are the ATCA-45xx series and the ATCA-46xx series CPMs, which only operate in enhanced mode.

To change the operating mode:

1. Establish a serial connection to the module and log on as root. See the *Installation Guide* for details.
2. To change to enhanced mode, enter:
`rsys-ipmitool rsysoem shmc enhanced <address>`
For example, to enable an ATCA-7220 in slot 1 of an ATCA-6002 shelf:
`rsys-ipmitool rsysoem shmc enhanced 0x82`
3. To verify that the module's Shelf Manager is enabled in enhanced mode, enter:
`rsys-ipmitool rsysoem shmc info`
If the Shelf Manager is enabled in enhanced mode, the response is ShMC state: Enhanced mode
4. To change back to basic mode, follow these steps again and enter this command instead:
`rsys-ipmitool rsysoem shmc basic <address>`

Overview of Shelf Manager and HPI configuration

Shelf Manager and HPI configuration settings can be either specific to the module running the Shelf Manager or specific to the shelf.

- *Shelf Manager-specific settings* affect how an individual module running the Shelf Manager performs its processing. These settings are stored on the module in the `/etc/shmgr.conf` file.

All configuration settings described in this chapter are specific to a Shelf Manager module. When the shelf has redundant SCMs, changes to these settings require editing the `/etc/shmgr.conf` file on both the active and standby SCM. See [Editing the `shmgr.conf` configuration file on page 42](#) for the recommended procedure.

- *Shelf-specific settings* affect the shelf or the FRUs in the shelf. You make changes to these settings on the active Shelf Manager module. Shelf-specific settings are described in [Chapter 4, Changing Shelf Settings Using HPI, on page 53](#).

[Table 4 on page 41](#) lists the Shelf Manager-specific settings you can configure by editing the `/etc/shmgr.conf` file.

Table 4. Shelf Manager and HPI configuration settings

Shelf Manager configuration settings	Parameter to edit in /etc/shmgr.conf file	For details, see:
Configuring verbosity and log settings	SHMS_CMDLINE_ARGS	page 43
Disabling the HPI service	SHMS_CMDLINE_ARGS	page 44
Configuring RMCP	SHMS_RMCP_INTERFACE SHMS_LOCAL_RMCP_PORT	page 44
Keeping Shelf Manager processes running	SHMS_KEEPAIVE	page 44
Configuring the system event log (SEL)	SHMS_SYNC_SEL_TIME SHMS_SEL_AUTOCLEAR	page 45
Setting the FRU heartbeat frequency	SHMS_FRU_HEARTBEAT_FREQUENCY	page 45
Setting the OpenIPMI channel number	SHMS_OPENIPMI_CHANNEL_NUMBER	page 45
Setting the minimum fan speed	SHMS_FAN_FLOOR	page 45
Setting the HPI redundancy mode	HSD_SHM_NON_REDUNDANT	page 46
Configuring HPI alarms	HSD_ALARM_MANAGER_ENABLE HSD_DAT_DISABLE HSD_ALARM_SITE_TYPE HSD_MAX_USER_ALARM_ENTRIES HSD_CHASSIS_FRU_MONITOR_ENABLE HSD_FAN_RPM_ALARMS	page 46
Configuring HPI multitasking	HSD_MAX_SIMULTANEOUS_REQUESTS	page 47
Configuring the shelf entity path	HSD_ROOT_ENTITY_PATH HSD_SHELF_ENTITY_LOCATION	page 47
Assigning an entity type for a new FRU	HSD_FRU_ENTITY_TYPE	page 48
Configuring HPI domain event log settings	HSD_MAX_EVENT_LOG_ENTRIES HSD_EVENT_LOG_OVERFLOW_ACTION HSD_USER_EVENT_MAX_SIZE	page 48
Setting the persistent storage location for HPI configuration files	HSD_PERSISTENT_CONFIG_HOME	page 48
Loading saved HPI resource configuration files	HSD_LOAD_SAVED_CONFIGURATION	page 49
Setting the standby HPI server to read-only	HSD_STANDBY_READ_ONLY	page 49
Setting IDRs to read-only	HSD_IDR_READ_ONLY	page 49
Loading RDRs from a module's HSD	HSD_LOAD_BLADEHSD_IPMI_RDRS	page 49

Editing the shmgr.conf configuration file

File-based configuration settings are stored in the */etc/shmgr.conf* file for each Shelf Manager module and are saved permanently. Any changes you make take effect when the Shelf Manager or the module running the Shelf Manager is restarted.

On an ATCA-6002 shelf, the */etc/shmgr.conf* file is supported only when the Shelf Manager is operating in enhanced mode. When the Shelf Manager is operating in basic mode, the */etc/shmgr_basic.conf* file is used instead. For details, refer to [Configuring the Shelf Manager in basic mode on page 50](#).

Note: The contents of your */etc/shmgr.conf* file override the configuration parameter default settings in */usr/share/shmgr/shmgr.defs*. However, you should not edit the *shmgr.defs* file, because the factory defaults should not be changed. All custom configuration should be done in */etc/shmgr.conf*.

To change the Shelf Manager and HPI configuration by editing the */etc/shmgr.conf* file:

1. On the Shelf Manager module, open the */etc/shmgr.conf* file in a text editor. (The *vi* text editor resides on the module's file system.)

If the shelf has redundant SCMs, open this file on the SCM containing the active Shelf Manager.

2. Edit the configuration parameters in */etc/shmgr.conf*, adding parameters as needed. When adding parameters, copy the syntax found in the defaults file */usr/share/shmgr/shmgr.defs*.
3. Save your changes to */etc/shmgr.conf*.
4. If the shelf has redundant SCMs, access the CLI on the SCM containing the active Shelf Manager and enter these commands:

```
shelf-mgmt
configure sync_config
exit
```

Your changes to */etc/shmgr.conf* are synchronized to the SCM containing the standby Shelf Manager.

5. Restart the Shelf Manager module (or the SCM containing the active Shelf Manager) to make your configuration changes take effect:

```
shmgr restart
```

6. Restart the SCM containing the standby Shelf Manager.

Configuring Shelf Manager behavior

This section describes the shelf management configuration settings you can change by editing shelf management server parameters (parameters with the prefix *SHMS*) in the */etc/shmgr.conf* file.

Configuring verbosity and log settings

The *SHMS_CMDLINE_ARGS* parameter in */etc/shmgr.conf* is where you can set verbosity and log settings. *SHMS_CMDLINE_ARGS* sets the command-line parameters passed to the Shelf Manager program when it starts. You can use the following values:

- `--verbosity [level]` The default is 5. Use 3 for a less verbose but informative debug level, 6 for a debug-friendly level, or 7 for the most verbose debug messages. If you use a value less than 3, you do not get error or information messages.
- `--maxlog [log size]` The default size of the */tmp/shmgr/shmgr.log* file is 8000000 (8 MB). You can increase the log size, but keep it below 16 MB. Logs are stored in RAM and using more than 16 MBs could make the system unstable.

Use higher values, up to 16000000 (16 MB), in order to log more messages. When the initial log file reaches the maximum size, it is compressed with *gzip* and copied to a backup file (*shmgr.log.1.tgz*). Then a new log is created. The second time the log reaches the size limit, it is copied to another backup file (*shmgr.log.2.tgz*). The first and second backups contain the initial startup information and are kept until the Shelf Manager module is restarted. The most recent ten backup files are also kept until a restart.

Note: The space required for the log file and its backups is roughly double the amount set in this parameter.
- `--nolog [hsd, shms]` Disables logging of HPI or ShMS messages.

For maximum debugging, the following settings are recommended:

```
SHMS_CMDLINE_ARGS="--verbosity 7 --maxlog 8000000"
```

Disabling the HPI service

The `SHMS_CMDLINE_ARGS` parameter in `/etc/shmgr.conf` lets you disable HPI service. `SHMS_CMDLINE_ARGS` sets the ShMS-related command-line parameters passed to the Shelf Manager program when it starts.

`--nohpi` Disables the HPI service, which is not recommended except in limited debugging situations. Disabling HPI causes the platform-management CLI to be unavailable and prevents the use of HPI applications.

If you decide to set this option, add it to the `SHMS_CMDLINE_ARGS` options already specified.

Configuring RMCP

The `SHMS_RMCP_INTERFACE` parameter in `/etc/shmgr.conf` specifies the default RMCP-enabled interface on the Shelf Manager. On the ATCA-2210, the default interface is `dtl0`.

The `SHMS_LOCAL_RMCP_PORT` parameter in `/etc/shmgr.conf` specifies the RMCP port for the local channel. This setting applies to the static IP address assigned to the module, and not the dynamic shelf IP address assigned to the active Shelf Manager.

WARNING! Do not change this setting except to satisfy a specific requirement. It is recommended that you check with Radisys Technical Support before making changes using these controls.

Keeping Shelf Manager processes running

The `SHMS_KEEPAWAKE` parameter in `/etc/shmgr.conf` enables the *keepalive* daemon that ensures that the Shelf Manager processes are continuously running.

Y Default. Enables the *keepalive* daemon.
N Disables this feature.

Configuring the system event log (SEL)

The `SHMS_SYNC_SEL_TIME` parameter in `/etc/shmgr.conf` sets the system event log (SEL) time on all IPMCs to the time on the active Shelf Manager.

- Y Default. Synchronizes the SEL time with the active Shelf Manager time.
- N Disables this feature.

The `SHMS_SEL_AUTOCLEAR` parameter in `/etc/shmgr.conf` controls how the SEL is cleared. The SEL drops new events if the log is full.

- Y Ensures that the SEL gets cleared each time the Shelf Manager is restarted.
- N Default. The local SEL is not cleared automatically.

Setting the FRU heartbeat frequency

The `SHMS_FRU_HEARTBEAT_FREQUENCY` parameter lets you set the frequency at which the Shelf Manager pings intelligent FRUs to verify IPMB communication. Minimum is 2 seconds. Maximum is 60 seconds. Default is 3 seconds.

Setting the OpenIPMI channel number

The `SHMS_OPENIPMI_CHANNEL_NUMBER` parameter enables you to set the channel number for sending requests when using the OpenIPMI driver. The default is 1. This parameter has no effect if the Radisys IPMI driver is in use.

WARNING! Do not change this setting except to satisfy a specific requirement. It is recommended that you check with Radisys Technical Support before making changes using these controls.

Setting the minimum fan speed

The parameter in `/etc/shmgr.conf` controls specifies the minimum fan speed as a percentage. The Shelf Manager never sets the fans below this speed. The default is 30.

Each type of shelf has a discrete number of possible fan speeds as identified in [Overriding the cooling algorithm on page 74](#). The percentage you specify is interpreted as the next lowest fan speed.

For example, on a shelf with 10 possible speeds, you can achieve a speed of 4 (40%) by specifying a percentage from 40 to 49.

Configuring HPI behavior

This section describes the HPI configuration settings you can change by editing HPI server daemon parameters (parameters with the prefix *HSD*) in the */etc/shmgr.conf* file.

Setting the HPI redundancy mode

The *HSD_SHM_NON_REDUNDANT* parameter in */etc/shmgr.conf* lets you specify whether HPI operates non-redundantly (without a second SCM installed).

- | | |
|---|--|
| Y | Enables non-redundant HPI operation. Do not enable this parameter unless the shelf is to be operated with only one SCM installed or if the shelf is the ATCA-6002. Setting this parameter to Y disables the HPI non-redundancy alarm and has no effect on the Shelf Manager. |
| N | Default. HPI operates redundantly. |

Configuring HPI alarms

The following parameters in */etc/shmgr.conf* let you configure HPI alarms:

- *HSD_ALARM_MANAGER_ENABLE* enables alarms to be communicated to the alarm manager module. On Radisys shelves, this includes the telco alarm contacts and the shelf display panel (SDP) on the ATCA-6000 shelf.

Y	Default. Enables the alarm module. In addition, <i>HSD_DAT_DISABLE</i> must be set to “n”, and <i>HSD_ALARM_SITE_TYPE</i> must be set to a valid value for the alarm manager to be enabled.
N	Disables the alarm module. This can be helpful in a case where the shelf doesn’t have an SDP installed, as in the case of the ATCA-6002 shelf.
- *HSD_DAT_DISABLE* disables the HPI domain alarm table (DAT), which also disables the HPI alarm module. The DAT contains the current fault conditions. Disabling it prevents new alarms from being added to it, and renders the alarm module useless.

Y	Disables the DAT, which also disables the alarm module. This is not recommended, except in debugging situations.
N	Default. Enables the domain alarm table.
- *HSD_ALARM_SITE_TYPE* sets the Site Type for the alarm module in hexadecimal. The default is 0xC1, which identifies the SDP. The alarm module will be disabled if the site type is set to 0xFF, or if the site type is not found in shelf address table.
- *HSD_MAX_USER_ALARM_ENTRIES* sets the maximum number of alarms that users can manually add to the DAT. The default is 50, and the maximum is 512.

- `HSD_CHASSIS_FRU_MONITOR_ENABLE` enables the monitoring of shelf FRUs, including the alarm module and fans. If monitoring is enabled, the shelf FRUs are scanned for their presence or removal from the shelf and appropriate actions are taken.

Y	Default. Enables monitoring of the shelf FRUs.
N	Disables shelf FRU monitoring.
- `HSD_FAN_RPM_ALARMS` enables fan RPM sensor events to be logged as alarms.

Y	Default. Enables logging of fan RPM sensor events as alarms.
N	Fan RPM sensor events are not logged as alarms.

Configuring HPI multitasking

The `HSD_MAX_SIMULTANEOUS_REQUESTS` parameter in `/etc/shmgr.conf` lets you configure HPI multitasking by setting the number of simultaneous HPI requests allowed across all sessions. HPI reserves a fixed number of threads based on this parameter. The default is 6, and the range of values is from 1 to 15.

Configuring the shelf entity path

The `HSD_ROOT_ENTITY_PATH` and `HSD_SHELF_ENTITY_LOCATION` parameters in `/etc/shmgr.conf` let you configure the shelf entity path. The shelf entity path is a string of location identifiers that uniquely identifies the physical shelf.

- `HSD_ROOT_ENTITY_PATH` sets the custom root entity portion of all entity paths. The root entity is appended to the end of all resource and RDR entity paths. The default is `"SAHPI_ENT_ROOT,0"`.

For example, to set entity paths to end with subrack 5, rack 10:

```
HSD_ROOT_ENTITY_PATH="SAHPI_ENT_SUBRACK,5:SAHPI_ENT_RACK,10:SAHPI_ENT_ROOT,0"
```

Valid separators between each entity element are: `:` `;` `{}` `()` `[]`

- `HSD_SHELF_ENTITY_LOCATION` sets a user-defined chassis number to be used as the entity location of the logical shelf resource. If set to the string `"shelf address"` or left undefined, the least significant four bytes of the shelf address are used as the shelf resource's entity location. The default is `"shelf address"`.

Assigning an entity type for a new FRU

The `HSD_FRU_ENTITY_TYPE` parameter in `/etc/shmgr.conf` lets you assign an entity type to a new FRU. To assign an entity type, use the following syntax:

```
HSD_FRU_ENTITY_TYPE=<Manufacturer Id>:<Product Id>:<HPI Entity Type in string format>
```

You must add a new entry in the configuration file for each FRU whose entity type you want to configure. Valid delimiters between each entry are: `:` `;` `{}` `()` `[]`

For example, to assign a FRU an entity type of `SAHPI_ENT_SBC_BOARD` with a manufacturer ID of 1234 and a product ID of 5678, you would add the following to the `shmgr.conf` file:

```
HSD_FRU_ENTITY_TYPE=1234:5678:SAHPI_ENT_SBC_BOARD
```

By default, the Shelf Manager detects and assigns entity types to certain types of Radisys FRUs. When the Shelf Manager encounters an unknown FRU in a front slot, it assigns a default PICMG entity type of `ATCAHPI_ENT_PICMG_FRONT_BLADE`. You can retrieve manufacturer and product ID by issuing the IPMI Get Device ID request as shown in [Table 10 on page 88](#).

Configuring HPI domain event log settings

The following parameters in `/etc/shmgr.conf` let you configure the HPI domain and resource event logs:

- `HSD_MAX_EVENT_LOG_ENTRIES` sets the maximum number of entries that can be held by the HPI domain event log or resource event log. The default is 512. The maximum is 1024.
- `HSD_EVENT_LOG_OVERFLOW_ACTION` specifies how new events are handled when an HPI event log is full.

“wrap” Wraps in FIFO mode by deleting the oldest event to make room for the newest event.

“drop” Drops new events when the log is full.

- `HSD_USER_EVENT_MAX_SIZE` specifies the maximum length of events manually written by the user. The default is 255 characters, which is the maximum.

Setting the persistent storage location for HPI configuration files

The `HSD_PERSISTENT_CONFIG_HOME` parameter in `/etc/shmgr.conf` lets you set the persistent storage location of HPI parameter control configuration files for all resources that have HPI parameter control capability. The default location is `/var/lib/shmgr`.

You can set this parameter to persistently store HPI configuration on a remote NFS-mounted server. This enables multiple shelves with similar configurations to use a common storage archive and load common configuration parameters.

For details on using this parameter, see [Saving and loading custom FRU configurations on page 76](#).

Loading saved HPI resource configuration files

The `HSD_LOAD_SAVED_CONFIGURATION` parameter in `/etc/shmgr.conf` enables the loading of previously saved HPI resource configuration files during Shelf Manager initialization.

- Y Enables the loading of saved HPI resource configuration files. This may result in increased startup time.
- N Default. Custom configuration files are not loaded at startup. To load these files manually for each resource, use the HPI `SaHpiParmControl` (`SAHPI_RESTORE_PARM`). For more information, see [Saving and loading custom FRU configurations on page 76](#).

Setting the standby HPI server to read-only

The `HSD_STANDBY_READ_ONLY` parameter enables you to prevent writing data to the standby HPI server. This helps the system manager to avoid conflicts and synchronization problems between the peer HPI servers if they are being accessed independently.

- Y Limits HPI users to read-only operations on the standby HPI server.
- N Default. Allows HPI users to perform read-write operations on the standby HPI server.

Setting IDRs to read-only

The `HSD_IDR_READ_ONLY` parameter enables you to prevent writing data to all inventory data records (IDRs) and their associated areas and fields.

- Y Limits HPI users to read-only operations on IDRs.
- N Default. Allows HPI users to perform read-write operations on IDRs.

Loading RDRs from a module's HSD

The `HSD_LOAD_BLADEHSD_IPMI_RDRS` parameter enables the loading of IPMI-based resource data records (RDRs) from a module's HSD. These RDRs can be used for debugging.

- Y Enables the loading of RDRs from the module's HSD resources.
- N Default. Disables this feature.

Configuring the Shelf Manager in basic mode

For a Shelf Manager running on the ATCA-6002 shelf, a configuration file for basic mode is located at */etc/shmgr_basic.conf*. When basic mode is enabled, the shelf management *init* script loads this file, which specifies the supported platform configurations. Usually this file does not need to be modified unless you need to install a replacement module.

The following options are available in *shmgr_basic.conf*.

Note: All options are case sensitive. There must be a space before and after the equal sign (=).

LOG_FILE	Configures where the script logs its output. By default, the location is set to LOG_FILE = /tmp/shmgr/shmgr_basic.log.
DEBUG_LEVEL	Specifies the level of verbosity. The value ranges from 0 to 7, with 7 being the most verbose. This option is usually commented out.
SLOT_ADDRESSES	A list of IPMB addresses for the shelf. For the ATCA-6002 shelf, this will always be configured as SLOT_ADDRESSES = 0x82 0x84.
POWERUP_RETRIES	Configures the number of seconds the script will attempt to power up all FRUs. The value ranges from 10 to 60, with 20 being the default.
EKEY_CONFIG	<p>Specifies which E-Keys should be enabled depending on which modules are in the shelf. Module types are specified on the same line as EKEY_CONFIG, in the following form:</p> <pre><Slot Address in hex 0xXX>-<Manufacturer ID in decimal>- <Product ID in decimal>,</pre> <p>Manufacturer ID and Product ID are values returned by the Get Device ID command.</p> <p>For example, the option for ATCA-4500 CPMs in both slots 1 and 2 is:</p> <pre>EKEY_CONFIG 0x82-4337-14451 0x84-4337-14451</pre> <p>Additional E-Key details may follow. For more information, refer to the comments in the configuration file.</p>

Editing the rsyshsd.conf configuration file

For each front module there are blade HPI configuration settings that are stored in the */etc/rsyshsd.conf* file and saved permanently. The default configuration settings work for most of the front modules without any modifications. For CPMs, however, the `RSYSHSD_BASE_INTERFACE` parameter must be edited to specify the name of the bonded interface. For details, refer to [Configuring blade HSD interfaces on page 52](#).

Any changes you make take effect when the module is restarted.

Note: The contents of the */etc/rsyshsd.conf* file override the configuration parameter default settings in */usr/share/shmgr/shmgr.defs*. Do not edit the *shmgr.defs* files directly. All custom configurations should be done in */etc/rsyshsd.conf*.

To change the blade HPI configuration:

1. On the front module, open the appropriate */etc/rsyshsd.conf* file in a text editor. (The vi text editor resides on the module's file system.)
2. Edit the configuration parameters, adding parameters as needed. When adding parameters, copy the syntax found in the defaults file */usr/share/shmgr/shmgr.defs*.
3. Save your changes.
4. Restart the front module to make your configuration changes take effect.

Configuring blade HPI behavior

This section describes the blade HPI configuration settings you can change by editing blade HPI server daemon parameters (parameters with the prefix *RSYSHSD*) in the */etc/shmgr.rsyshsd.conf* file.

Disabling the blade HPI server

The `RSYSHSD_DISABLE` parameter allows you to disable the blade HPI server. This parameter is normally only used for debug purposes.

- | | |
|---|------------------------------------|
| Y | Disables the blade HSD. |
| N | Default. The blade HSD is enabled. |

Configuring log settings

- The `RSYSHSD_LOG_FILE` parameter is where you can specify the name of the log file. If you do not specify a value, the software chooses a suitable name.
- The `RSYSHSD_CMDLINE_ARGS` parameter is where you can set verbosity and log settings. You can use the following values:
 - `--verbosity [level]` The default is 5. Use 3 for a less verbose but informative debug level, 6 for a debug-friendly level, or 7 for the most verbose debug messages. If you use a value less than 3, you do not get error or information messages.
 - `--maxlog [log size]` The default size of the log file is 8000000 (8 MB). You can increase the log size, but the value must be less than half of the amount of free disk space available.

Configuring blade HSD interfaces

The `RSYSHSD_BASE_INTERFACE` and `RSYSHSD_FABRIC_INTERFACE` parameters specify the default Base and Fabric interfaces for the blade HSD. The shelf HPI server listens for HPI session open requests on these interfaces.

For example:

```
RSYSHSD_BASE_INTERFACE=dt10:2
RSYSHSD_FABRIC_INTERFACE=dt11:2
```

The `RSYSHSD_SHMS_INTERFACE` parameter specifies the interface to be used for communication with the Shelf Manager. This parameter must be set to either the Base or Fabric interface as defined by `RSYSHSD_BASE_INTERFACE` and `RSYSHSD_FABRIC_INTERFACE`.

For example:

```
RSYSHSD_SHMS_INTERFACE=dt10:2
```

Important: For the ATCA-45xx series and the ATCA-46xx series CPMs, the recommended way to set up the blade HSD is to specify the name of the bonded interface for the `RSYSHSD_BASE_INTERFACE` parameter. This ensures that the blade HSD server can always maintain connectivity to an active Shelf Manager, regardless of failovers.

Keeping blade HPI server processes running

The `RSYSHSD_KEEPLIVE` parameter enables the *keepalive* daemon that ensures that the blade HPI server processes are continuously running.

- | | |
|---|---|
| Y | Default. Enables the <i>keepalive</i> daemon. |
| N | Disables this feature. |

Changing Shelf Settings Using HPI

This chapter describes the Radisys HPI implementation, including how HPI affects your applications, how to write HPI applications, and how to change various shelf settings. You can change the shelf settings using any of these methods:

- Platform-management CLI commands
- Standard PICMG commands
- Standard HPI controls
- SNMP requests to operate on HPI-B0101-MIB module objects
- *hpiapp* menu options

Note: The settings described in this chapter are *shelf-specific*. Shelf-specific settings affect the shelf or FRUs on the shelf. You make changes to these settings on the Shelf Manager module. Shelf-specific settings are stored in the shelf FRU device, so if there is a redundant SCM, you do not need to copy them there.

Radisys HPI implementation details

The HPI Open Session function, *saHpiSessionOpen()*, supports an option for specifying an implementation-specific default domain. Since the Radisys implementation allows the client to run remotely, Radisys explicitly specifies the default domain on the client.

The Radisys HPI client library (HCL) uses an environment variable (SAHPI_UNSPECIFIED_DOMAIN_ID) to specify the default domain ID. If an application uses this environment variable, it must be set to the IP address of the active Shelf Manager prior to using the library in the open session function.

The Radisys HCL, provided as a shared library, implements and responds to all functions defined in the *SAF HPI Specification*. The HPI domain, resource, entity, and management instrument architecture is compliant with the *SAF Mapping Specification*.

The mapping specification describes most of the resources and associated management instruments provided by the Radisys HPI implementation. [Table 5 on page 54](#) lists Radisys-specific resources not found in the mapping specification.

Table 5. Radisys-specific resources

Resource type ¹	Container resource	Description	Entity (type, location) ²	Lifetime ³
Shelf	Root	This logical resource represents the shelf. There is exactly one created by each HPI server. This resource is implemented as described in the mapping specification. However, the format of the entity instance of the shelf resource is Radisys-specific.	(SAHPI_ENT_ADVANCEDTCA_CHASSIS, Shelf Id) The entity instance of the shelf resource is the first 4 bytes of the shelf address, and is an encoding of the physical location of the shelf.	Static
SPM slot	Shelf	This logical resource represents the slot occupied by a shelf peripheral module (SPM).	(SAHPI_ENT_OEM_SPECIFIC, SPM Site Number)	Static
Virtual SPM slot	Shelf	This logical resource provides support for the logical virtual SPM in the shelf. The virtual SPM represents the functionality accessible only through the active SPM.	(SAHPI_ENT_OEM_SPECIFIC+2, 1)	Static
SPM	SPM slot	This physical resource represents the SPM.	(SAHPI_ENT_IO_BLADE, Site Number of SPM)	Dynamic
COM-E Slot	SCM FRU resource	This logical resource represents the COM-E module slot carried by the ATCA-2210.	(SAHPI_ENT_OEM_SPECIFIC+3, 1)	Dynamic
COM-E Module	COM-E slot	This physical resource represents the COM-E module that can be installed on an ATCA-2210.	(SAHPI_ENT_ADD_IN_CARD, COM-E slot site number)	Dynamic
RCM slot	Shelf	This physical resource represents the slot occupied by the RCM.	(SAHPI_ENT_OEM_SPECIFIC, RCM Site Number)	Static
RCM	RCM slot	This physical resource represents the RCM.	(SAHPI_ENT_IO_BLADE, RCM Site Number)	Dynamic

Notes:

1. The resource types listed in this table may be different depending on the applicable modules. For example, the ATCA-7220 PPM does not have a COM-E slot or an SPM.
2. The site number used for the entity location refers to the site number associated with the physical slot listed in the AdvancedTCA address table contained in the shelf FRU information.
3. The Lifetime column specifies when the resource is added or removed from the resource presence table (RPT). Static resources are created on power-up and are never removed from the RPT. Dynamic resources are created when the resource is detected as present and deleted from the RPT when the resource is detected as absent.

The DomainId needed for opening HPI sessions on the Radisys platform's HPI Interface is derived from the Shelf Manager's IP address. See [Writing HPI applications on page 57](#) for more information on deriving the DomainId parameter from IPv4 addresses.

The resource identifier parameter, ResourceId, for any resource in the Radisys platform's HPI Interface is formatted for easy preliminary identification of the entity being represented by the resource.

The following is the ResourceId format used by shelf and Shelf Manager resources in the HPI resource presence table (RPT):

- 0x01 Shelf resource. This represents the entire shelf and provides access to some shelf-related configuration such as the shelf address, Shelf Manager IP address, and so forth. This resource contains an inventory data record that also provides access to the shelf FRU information.
- 0x02 Shelf Manager resource. This represents the Shelf Manager and all functionality associated with the dedicated active Shelf Manager at IPMB address 0x20.

The following is the ResourceId format used by FRU and slot resources in the RPT:

XXYYZZ

XX, YY, and ZZ each represent one byte in hexadecimal format, thus making the ResourceId 3 bytes long.

- XX Resource location. This is the location or instance number of the site, such as the physical slot number, associated with the entity represented by the resource. For AMCs, this is the AMC site number on a carrier. For non-intelligent FRUs, it is the device-specific site number on the proxy FRU.
- YY Resource type. This indicates the type of the entity being represented by the resource. For a FRU, this is the ATCA Site Type associated with the slot in which the FRU is installed. For debugging purposes, the Radisys resource types are identified in [Resource types of slots and FRUs in the RPT on page 135](#).
- ZZ IPMB address of the entity represented by the resource. For details, see [IPMB addresses of slots and FRUs on page 134](#). For AMCs and non-intelligent FRUs, this is the IPMB address of the carrier or proxy FRU, respectively.

Important: The ResourceId should **not** be used as the primary identification mechanism. HPI applications should use the entity path of a resource to properly identify its type and location. For details on entity path construction, see the *SAF Mapping Specification*.

Using the platform-management CLI

You can access several command modes, including platform-management mode, from the Shelf Manager module's master (top-level) CLI. The platform-management mode provides a CLI you can use to configure the Shelf Manager, control the Shelf Manager service, view and acknowledge all alarms in the shelf, and view information about the FRUs in the shelf.

For information about specific CLI commands, see the *CLI Reference*.

Accessing the platform-management CLI

To access platform-management mode and the platform-management CLI, you must first access the active Shelf Manager module's master CLI:

1. Access the module hosting the active Shelf Manager using Telnet or the serial console.
2. Log on to the Shelf Manager module as an administrator:

admin

The Shelf Manager module's master CLI prompt appears. For example:

ATCA-2xxx#

3. Access the platform-management CLI by entering:

platform-mgmt

From the platform-management CLI, you can access several command modes. To see a list of these command modes, enter:

?

To exit the platform-management CLI or any of the command modes, type:

exit

See the *CLI Reference* for descriptions of the platform-management command modes and the commands.

Writing HPI applications

The following are required to create an HPI application:

- | | |
|--------------------|---|
| <i>SaHpi.h</i> | The official HPI B.03.02 header file, which is required to create any HPI B.03.02 user application. The header file is also backward compatible with HPI B.01.01 and HPI B.02.01. You can download this from the Service Availability Forum (SAF) Web site (http://www.saforum.org), and it is also included as part of the Radisys HPI client library and application package. |
| <i>SaHpiAtca.h</i> | The AdvancedTCA-specific header file, which provides definitions and symbols for all standard resources and management instruments modeling different aspects and entities of an AdvancedTCA platform. Although <i>SaHpiAtca.h</i> is not necessary to create a standard HPI application, an AdvancedTCA-specific HPI application can use the symbols defined in this file to directly call out AdvancedTCA-specific resources, sensors, and controls. This file is included as part of the Radisys HPI client library and application package, and can also be downloaded from the SAF Web site. |
| <i>libhcl.so</i> | The HCL for Linux, which user applications need to link with in order to communicate with the HPI server. The HCL can be linked at compile time or loaded during runtime with an HPI-compliant application. The HCL is installed on the Shelf Manager module, and is also available on the Radisys FTP site as source code with build instructions for Linux operating systems. The archive also includes the <i>hpiapp</i> example application source code, along with other HPI demo applications. |

To run any HPI application that links with the Radisys HPI client library, *libhcl.so*, consider the following:

- The *SaHpi.h* header file and the HCL support HPI B.03.02 and are backward compatible with HPI B.02.01 and HPI B.01.01. Radisys recommends recompiling your HPI applications with the new header file.
- Ensure the run-time environment is properly set up so that applications can find the HPI client library. Do this by completing either of the following tasks:
 - Build and install the HPI client library, *libhcl.so*, and note the installation location. Build instructions are provided in the HPI client library and application package included on the Radisys FTP site.
 - Set the LD_LIBRARY_PATH environment variable to the location of the *libhcl.so*.

You must convert the IPv4 address to a binary format because the DomainId is defined as a 32-bit unsigned integer and would not be able to accept an ASCII-formatted IPv4 address. HPI applications can use the Linux library function *inet_addr()* to convert the IPv4-based Shelf Manager Address to a binary format.

- To open sessions to a remote HPI server using the default DomainId SAHPI_UNSPECIFIED_DOMAIN_ID, set the environment variable SAHPI_UNSPECIFIED_DOMAIN_ID to the IP address of the Shelf Manager. This should normally be the dedicated Shelf Manager IP address that is bound to the active Shelf Manager. For example, the following Bourne shell command creates and exports the environment variable SAHPI_UNSPECIFIED_DOMAIN_ID to the Shelf Manager IP address XX.XX.XX.XX.

```
export SAHPI_UNSPECIFIED_DOMAIN_ID=XX.XX.XX.XX
```

The HCL searches for this environment variable when a request to open a session using the default domain identifier is received. If this variable is not set, the HCL will use 127.0.0.1 to deduce the DomainId.

If no HPI server responds at the specified IP address, *saHpiSessionOpen()* will return the error code: SA_ERR_HPI_INVALID_DOMAIN.

- When running the HPI application remotely, it might be necessary to set up a new route for the Shelf Manager's IP address so that communication can be established between the remote machine hosting the HPI application and the HPI server.

Using OpenHPI tools

OpenHPI is an open source project created with the intent of providing an implementation of SAF's HPI. This open source community provides several HPI applications, plug-ins, and utilities that can be used with the Radisys HPI implementation. The Radisys HPI client library is fully compatible with all OpenHPI tools.

You can download the OpenHPI software suite from the OpenHPI Web site:

<http://www.openhpi.org/>

Refer to the README file that comes with the OpenHPI software for instructions on configuring and building the OpenHPI software suite.

Some of the useful OpenHPI applications are stored in the *clients* and *hpi_shell* directories of the OpenHPI software. Do the following to link these applications with the Radisys platform's HPI client library:

1. Download the latest OpenHPI software from <http://www.openhpi.org/>.
2. Expand the *tar.gz* archive. Configure and build the entire OpenHPI software as instructed in the included README file. You do not need to install the software suite to run it.
3. Change the directory to *src/.libs* directory. This is the location of the built OpenHPI libraries. *libopenhpi.so* is a symbolic link to the OpenHPI library for HPI B.03.02.

Note: If the OpenHPI software suite was installed using the *make install* option, the *libopenhpi.so* library was installed in the standard Linux library location */usr/local/lib*.

4. Copy the Radisys HPI client library *libhcl.so* to the directory containing the OpenHPI library.
5. Redirect the symbolic link *libopenhpi.so* to point to *libhcl.so* (*libopenhpi.so*→*libhcl.so*):

Note: The version number in the following code is dependent on the OpenHPI version that you are using; for example, *libopenhpi.so.2.2.1*.

```
ln -sf libhcl.so libopenhpi.so.<version number>
```

6. Change to the OpenHPI clients directory (*../clients*). Any of the OpenHPI applications residing here, such as *hpitree*, *hpievent*, and *hpiensor*, can now be run with the Radisys platform's HPI client library.

Ensure SAHPI_UNSPECIFIED_DOMAIN_ID is set as an environment variable with its value being that of the Shelf Manager IP address of the shelf to be managed. See [Writing HPI applications on page 57](#) for more information on setting the DomainId.

The OpenHPI software suite is a source of useful HPI sample applications, plug-ins, and utilities. Radisys encourages downloading and customizing these applications for specific function and management requirements.

Using SNMP

SNMP access to the shelf management functionality is provided through the HPI-B0101-MIB module, which works seamlessly with the B.02.01 HPI. The HPI subagent on each Shelf Manager module facilitates SNMP management of the ATCA shelf by populating the objects and generating the notifications defined in the HPI-B0101-MIB module. On each Shelf Manager module the HPI subagent communicates with the HPI server using the HPI client library functions.

Note: The HPI subagent is not supported on the ATCA-6002 shelf.

The columnar objects in each HPI-B0101-MIB module table derive their instance values from one or more index objects. The Radisys HPI subagent assigns index values in these ways:

- Most tables use saHpiDomainId as the first index object which is always represented by the SAHPI_UNSPECIFIED_DOMAIN_ID value of 4294967295 (0xffffffff).
- Tables associated with HPI resources are also indexed by the saHpiResourceId object. One exception is the saHpiResourceTable itself, which is indexed by saHpiResourceEntryId.

Important: The SNMP management application should never assume specific values for the saHpiResourceId and the saHpiResourceEntryId objects. Instead the SNMP management application should determine the values by reading the saHpiResourceTable and associating the saHpiResourceId and saHpiResourceEntryId values for each resource with its saHpiResourceEntityPath value, which identifies the resource's function and hierarchical location within the shelf.

In addition, the SNMP management application should not assume any numerical relationship between the saHpiResourceId and the saHpiResourceEntryId values. The HPI subagent currently uses the same value for both for each resource, but this may change in the future.

- The HPI subagent populates only the objects associated with resources currently present in the physical shelf. After a resource is removed, the objects representing the resource are not preserved. Thus, the value of the saHpiResourceIsHistorical index object, which is also used to index many tables, is always false(2).

For each HPI control, the HPI subagent populates a row in one of six different tables:

```
saHpiCtrlAnalogTable
saHpiCtrlDigitalTable
saHpiCtrlDiscreteTable
saHpiCtrlOemTable
saHpiCtrlStreamTable
saHpiCtrlTextTable
```

Each of these tables is indexed by:

```
saHpiDomainId,  
saHpiResourceId  
saHpiResourceIsHistorical  
saHpiCtrl*EntryId object
```

The control number is used as the value for each instance of the saHpiCtrl*EntryId object. Thus, the value for the saHpiCtrl*EntryId variable in each table row is the same as the value for the saHpiCtrl*Num variable.

For a list of the tables and notifications that the HPI subagent supports, see the *Software Guide*.

Enabling the HPI subagent

You must configure and enable the SNMP agent before you enable the HPI subagent. See the *Software Guide* for details.

Note: The HPI subagent is not supported on the ATCA-6002 shelf.

To enable the HPI subagent, access the CLI on the Shelf Manager module and enter:

```
blade-mgmt  
config  
service hpiSubagent
```

To disable the HPI subagent, enter:

```
blade-mgmt  
config  
no service hpiSubagent
```

The Radisys software supplies an SNMP watchdog to monitor and restart the SNMP agent and the HPI subagent. For details on configuring and enabling the SNMP watchdog, see the *Software Guide*.

Configuring HPI subagent logging

HPI subagent logging is disabled by default. To enable logging:

1. In the `/etc/rc.d/init.d/hpiSubagent` startup script, set the desired type of logging:
 - To set logging to `/var/log/messages` via syslog, change the `OPTIONS` setting from `-Ln` to `-Lsd`
 - Alternatively, to set logging to `/var/log/hpiSubagent.log`, change the `OPTIONS` setting from `-L` to `-Lf /var/log/hpiSubagent.log`.
2. To enable the syslog daemon, access the CLI on the Shelf Manager module and enter these commands:

```
blade-mgmt
config
service syslog
```

Using the example HPI application (hpiapp)

Radisys provides an example HPI application, *hpiapp*, that is compliant with the *SAF HPI Specification*. You can use it to perform tasks and as an example when writing your own HPI applications.

Important:

- Do not call the *hpiapp* example application programmatically. It is an example only and is not intended for use in high availability applications.
- Do not use the *saHPIResourceResetStateSet()* function to reset AMCs installed in an ATCA-1200 module. The function uses the PICMG “FRU Control” command, which will perform a cold reset and cause the entire ATCA-1200 module to reset.

The following describes *hpiapp*:

- Pre-compiled binary is included in the module software image for the convenience of developers and provides menus for executing all HPI API calls defined in the HPI specification.
- Source code is provided in a tar format that you can decompress. Use the standard GNU auto-configure and auto-make processes to build it.
- Can be run either remotely or locally and links with the HPI client library (HCL) to interface with the HPI server running with the Shelf Manager.

A README file provided with *hpiapp* in the HCL source package explains in detail the build procedure and how to run it either locally or remotely. The *hpiapp* source provides an example of how to call most HPI functions.

Running hpiapp

The following examples show how to run the *hpiapp* example application to connect to an HPI server running Shelf Manager at the IP address of 10.100.22.248.

From the Linux prompt:

- Use the `-h` (host identifier) option:
`hpiapp -h 10.100.22.248`
or
- Set the default domain identifier to 10.100.22.248:
`export SAHPI_UNSPECIFIED_DOMAIN_ID=10.100.22.248`
`hpiapp`

When no `-h` option is present, *hpiapp* tries to use the default DomainId and automatically looks for the `SAHPI_UNSPECIFIED_DOMAIN_ID` environment variable.

hpiapp displays this menu of commands:

```
Commands:
0 - Get DomainInfo
1 - Set DomainTag
2 - Get Drt Entries
3 - Get Rpt Entries...
4 - Get Rdr Entries...
5 - Request Discover
a - Annunciator Commands...
c - Control Commands...
d - DAT (Alarm) Commands...
e - Event Commands
f - Fumi commands
h - Hotswap Commands...
i - IDR Commands...
m - Cnfg Mgmt Commands...
o - Toggle Output Mode (Summary)
p - ParmControl()
s - Sensors Commands...
w - Watchdog Commands...
? - Help
q - Quit
Cmd>
```

Select a command by typing a character.

Changing the shelf address

The first 4 bytes of the shelf address should be changed manually to uniquely identify the shelf according to its location.

Note: When changing the shelf address, do not deviate from the address format defined in [Radisys shelf address format](#) (below). Deviations from this format can cause the fruinfo fail alarm on the Shelf Manager module, which causes all modules to boot up in debug mode.

The address of a Radisys shelf is stored in the MultiRecord area of the shelf FRU device's address table. You can change the shelf address using any of the following methods:

- Using CLI commands. Enter:

```
platform-mgmt
controls
configure shelf-address numeric
```
- Using the standard PICMG commands Get/Set Shelf Address Info (see the *AdvancedTCA Base Specification*).
- Using the standard HPI shelf address control (control 0x1000) located in the shelf resource (ResourceId 0x01).
- Using the SNMP variables saHpiCtrlTextMode.0xffffffff.1.false.0x1000 and saHpiCtrlTextState.0xffffffff.1.false.0x1000 from the HPI-B0101-MIB module. First set the text control mode for the shelf address to manual(2). Then set the text control state to the new shelf address in hex. Finally, set the mode back to auto(1).
- Using the *hpiapp* example application interactively or as a programmatic example for writing your applications.

Radisys shelf address format

Of the 20 bytes that PICMG defines for the shelf address, the Radisys shelves use the first 6 bytes, as shown in [Table 6](#). The remaining 14 bytes are set to zeros by default. The DHCP server uses the first 6 bytes of the shelf address to allocate unique IP addresses across multiple shelves.

Table 6. Shelf address format

Field	Byte	Description
Vertical	0	Vertical position of the shelf in the rack
Horizontal	1	Horizontal position of the shelf in the rack
Row	2	Rack row location in the central office
Column	3	Rack column location in central office
Cluster rack	4	Logical rack in the cluster
Cluster shelf	5	Logical shelf in the cluster

Note: Bytes 4 and 5 of the shelf address are used to create the MAC identifier for the Base interconnect interface. If the shelf address is not configured following the Radisys format, the Base interconnect interface will not be configurable using DHCP.

Changing the Shelf Manager IP address

The active shelf management server has a dedicated IP address that is derived from the *Shelf IP Connection Record* in the shelf FRU device. The IP Connection Record stores:

- ShMS IP address (default is 192.168.16.17, or C0 A8 10 11 hexadecimal)
- Gateway address (default is 0.0.0.0)
- Subnet mask (default is 255.255.255.0, or FF FF FF 00 hexadecimal)

You can change the default ShMS IP address, gateway, and subnet mask:

- Using the CLI.

To change the default ShMs IP address, enter:

```
platform-mgmt
controls
configure shelf-ip-address
```

To change the gateway address, enter:

```
platform-mgmt
controls
configure shelf-gateway-address
```

To change the subnet mask:

```
platform-mgmt
controls
shelf-subnet-mask
```

- Using the standard HPI Shelf Manager IP address control (control 0x1001) located in the logical shelf resource (ResourceId 0x01). This is a Text control with three lines and supports only numeric data (Data Type — Binary). Each line supports a length of 4 bytes for the 4-byte IPv4 address.
- Using the SNMP variables saHpiCtrlTextMode.0xffffffff.1.false.0x1001 and saHpiCtrlTextState.0xffffffff.1.false.0x1001 from the HPI-B0101-MIB module. First set the text control mode for the Shelf Manager IP address to manual(2). Then set the text control state to the new IP address, gateway address, and subnet mask (each in hex). Finally, set the mode back to auto(1).
- Using the *hpiapp* example application interactively or as a programmatic example for writing your applications.

The record is divided into lines as follows:

- Line 1 – Shelf Manager IP address
- Line 2 – Gateway address
- Line 3 – Subnet mask

You can either change one line at a time or change all three lines together using the *saHpiControlSet()* function. Having the option of changing one line at a time allows you to update just one of the addresses (IP address, gateway or subnet mask) while leaving the other two untouched.

For details on the Shelf Manager IP address control, refer to the *SAF Mapping Specification*.

Configuring the FRU power-on sequence

The sequence by which FRUs are powered on during shelf power-on is dictated by the order in which their associated power descriptors are listed in the shelf activation and power management record in the shelf FRU device. Changing the FRU power-on sequence requires updating the ordering of the power descriptors in the shelf activation and power management record.

You can change the FRU power-on sequence:

- Using the CLI. See [Changing the FRU power-on sequence using the CLI on page 68](#) for instructions.
- Using the standard HPI controls (0x1300 to 0x1300+<#_of_slots>) located in the logical shelf resource (ResourceId 0x01). These digital controls are described in the sections that follow.

- Using the SNMP variables:

```
saHpiCtrlDiscreteMode.0xffffffff.1.<0x1300+sequence#>  
saHpiCtrlDiscreteState.0xffffffff.1.<0x1300+sequence#>  
saHpiCtrlDigitalState.0xffffffff.1.false.0x1300
```

from the HPI-B0101-MIB module.

First get the discrete control state values to determine the current power-on sequence by ResourceId, and ensure that the discrete control mode values are set to manual(2) for each ResourceId to be changed. Next set the new discrete control state values to change the power-on sequence. Finally, commit the new values by setting the digital control state to pulseOn(4).

For details on the FRU power-on sequence control, refer to the *SAF Mapping Specification*.

Changing the FRU power-on sequence using the CLI

You can use the CLI to:

- View the current power-on sequence
- Change the power-on sequence
- Commit the current power-on sequence (make it permanent).
- View the committed status sensors.

To view the current power-on sequence, enter:

```
platform-mgmt
controls
show [all | controlId]
```

To change the power-on-sequence, enter:

```
platform-mgmt
controls
configure fru-power-on-seq
```

To commit the current power-on sequence, enter:

```
platform-mgmt
controls
configure fru-power-on-seq-commit
```

To view a committed status sensor, enter:

```
platform-mgmt
sensors
show sensorId <number>
```

For additional information on the CLI commands, see the *CLI Reference*. For information about controls and sensors, see the sections that follow.

FRU power-on sequence controls

The shelf resource (ResourceId 0x01) provides a set of HPI discrete controls, each of which maps to a power descriptor. The discrete control state values map to the slot ResourceIds of each of the sites associated with the power descriptors.

Each intelligent FRU slot in the shelf has one power descriptor in the shelf activation and power management record in the shelf FRU information. A corresponding number of FRU power-on sequence controls exist in the shelf resource's RDR, which are numbered starting with 0x1301. The slot ResourceId listed as the control state of control 0x1301 is the first site to be powered on, the slot ResourceId listed as the control state of control 0x1302 is the next site to be powered on, and so on.

The HPI application can change the discrete state of any of these controls to any of the valid slot ResourceIds to modify the sequence in which the sites (represented by the slot resources) are powered up by the ShMS.

For example, in the ATCA-6006 shelf, the default power-on sequence has front slot 3 as 8th and slot 5 as 10th to receive power. To exchange the positions of these slots in the sequence, so slot 3 becomes 10th and slot 5 becomes 8th:

1. Change the discrete state value of control 0x130A to 0x3FF86 (slot ResourceId of front slot 3).
2. Change the discrete state value of control 0x1308 to 0x5FF90 (slot ResourceId of front slot 5).

FRU power-on sequence commit control

Changing the control states alone would not make the change in the shelf FRU information persistent. Digital control 0x1300 is used to initiate the process of reordering the power descriptors in the shelf FRU information. This digital control, known as the FRU power-on sequence commit control, accepts only the SAHPI_CTRL_STATE_PULSE_ON value and updates the shelf activation and power management record to the desired power-on sequence after successful completion. After updating the Power On Sequence control states for the discrete Power On Sequence controls (0x1301–0x1316) as desired, the HPI application must use control 0x1300 to do a ControlSet with digital value = SAHPI_CTRL_STATE_PULSE_ON to commit the changes to the shelf FRU information and the shelf FRU device, where it will be persistent following a shelf Power-On Reset (POR).

FRU power-on sequence commit status sensor

The shelf resource contains an additional sensor called the FRU power-on sequence commit status sensor (0x1300). This sensor indicates whether the configured power-on sequence has been committed. This sensor is an OEM sensor supporting only event states. No numeric readings or thresholds are supported. The following are the HPI sensor-specific event states supported by this sensor:

0x0001 (SAHPI_ES_STATE_00)	Status reported by the FRU power-on sequence controls is the present status in the shelf activation and power management record (that is, the power-on sequence in the shelf activation and power management record is up to date).
0x0002 (SAHPI_ES_STATE_01)	One or more of the FRU power-on sequence controls have been modified but not yet committed to the shelf FRU information. You must use the FRU power-on sequence commit control (control 0x1300) to write the updated power-on sequence to the shelf FRU information.

Error conditions

When doing a commit using the commit control, if duplicate ResourceIds are found in any of the 22 Power On Sequence controls, the error code SA_ERR_HPI_DUPLICATE is returned.

While setting the control state of any Power On Sequence control, if the ResourceId is not found to be a valid slot resource's ID, the error code SA_ERR_HPI_INVALID_DATA is returned.

Querying the current hot-swap state of a FRU

You can query a FRU's current hot-swap state:

- Using the CLI commands to get a summary of FRU information, including firmware versions and hot-swap states. Enter:

```
platform-mgmt
show frus
```

For information on additional FRUs, such as RTMs, fans, and power equipment modules, enter:

```
platform-mgmt
show chassis status
```

- Using the HPI hot-swap function *saHpiHotSwapStateGet()* to query the hot-swap state of the FRU associated with a resource.
- Using the SNMP variable *saHpiHotSwapState.0xffffffff.<FRU-ResourceId>.false* from the HPI-B0101-MIB module to get the hot-swap state of a particular FRU.
- Using the *hpiapp* example application interactively or as a programmatic example for writing your applications.

The ATCA FRU 'M' states are mapped to the HPI hot-swap states as follows:

Figure 8. M state to hot-swap state mapping

FRU "M" state	HPI hot-swap state
M0	Not Present
M1	Inactive
M2 and M3	Insertion Pending
M4	Active
M5 and M6	Extraction Pending
M7	Lost Communication

For more details on how the ATCA hot-swap state machine is mapped to the HPI hot-swap model, refer to the hot swap section of the *SAF Mapping Specification*.

Overriding the default ShMS hot-swap mechanism

Whether a FRU sitting in a particular slot is activated by the Shelf Manager or by the system manager is defined by the Shelf Manager controlled activation bit in byte 4 of the power descriptor associated with the hardware address of the slot. The power descriptors for all permanent slots associated with intelligent FRUs are present in the shelf activation and power management record of the shelf FRU information.

The default state of the Shelf Manager controlled activation bit in all the power descriptors is 1, meaning that, by default, the ShMS will activate a FRU upon reception of the M1-to-M2 hot-swap transition event from that FRU. The HPI application can change this policy by using the FRU activation control present in every slot resource's RDR table.

You can override the default hot-swap mechanism:

- Using the standard HPI FRU activation control (control 0x1020) located in the shelf resource (ResourceId 0x01). For details, see [HPI FRU activation control](#) (below).
- Using the following SNMP variables:

```
saHpiCtrlAnalogMode.0xffffffff.<slot-resourceID>.false.0x1020  
saHpiCtrlAnalogState.0xffffffff.<slot-resourceID>.false.0x1020
```

from the HPI-B0101-MIB module.

To keep a slot from being automatically activated, set the analog control mode for the slot to manual(2). To change the power-on-delay for the next FRU in the sequence, set the analog control mode to manual briefly, then set the analog control state for the slot to the new value in tenths of a second, then set the mode back to auto(1).

- Using the *hpiapp* example application interactively or as a programmatic example for writing your applications.

HPI FRU activation control

The FRU activation control is an analog control (0x1020) that can be used to set/clear the Shelf Manager controlled activation bit in the power descriptor associated with the hardware address of the slot resource whose RDR repository it resides in. This control can also be used to modify the Delay Before Next Power On value in tenths of a second for that hardware address. All slot resources will have the FRU activation control in their RDR repository. The FRU activation control supports both manual and auto modes:

- | | |
|--------|---|
| Manual | Clears the Shelf Manager controlled activation bit. A FRU inserted in the slot associated with this slot resource will not be activated by the Shelf Manager. The system manager will need to manually activate it using the standard HPI hot-swap management functions (refer to the managed hot swap section of the <i>SAF HPI Specification</i>). |
| Auto | Sets the Shelf Manager controlled activation bit. A FRU inserted in the slot associated with this slot resource will be activated by the Shelf Manager. |

While in manual mode, you can also update the Delay Before Next Power On for that hardware address. This is the analog state value of the control. New values are entered in tenths of a second. To change the delay before the next power on:

1. Briefly set the control in manual mode.
2. Change the analog state value of the control.
3. Reset the control to auto mode.

Refer to the FRU activation control section of the *SAF Mapping Specification* for more details on this control.

HPI FRU hot-swap policy cancellation

HPI provides another override feature for the system manager to stop Shelf Manager controlled activation of a FRU even if the associated FRU activation control is in auto mode.

For up to 3 seconds after the HPI hot-swap event for the FRU resource's transition to SAHPI_HS_STATE_INSERTION_PENDING is received, system managers can use the HPI API call *saHpiHotSwapPolicyCancel()* to abort FRU activation and hold the FRU in the M2 (Activation Request) hot-swap state. The system manager can later use the HPI API call *saHpiResourceActiveSet()* to initiate FRU Activation or *saHpiResourceInactiveSet()* to fully deactivate the FRU, as desired.

Similarly, for up to 3 seconds after the HPI hot-swap event for the FRU resource's transition to SAHPI_HS_STATE_EXTRACTION_PENDING is received, system managers can use the HPI API call *saHpiHotSwapPolicyCancel()* to abort FRU deactivation and hold the FRU in the M5 (Deactivation Request) hot-swap state. The system manager can later use the HPI API call *saHpiResourceActiveSet()* to take the FRU back to the active state or *saHpiResourceInactiveSet()* to initiate deactivation of the FRU, as desired.

The HPI hot-swap management functions, along with the FRU activation control, thus provide the ATCA Shelf Manager multiple methods to override the default Shelf Manager-controlled hot-swap insertion or extraction of intelligent FRUs from the shelf.

Overriding the cooling algorithm

System managers can completely override the Shelf Manager's cooling algorithm and use an external cooling module to control the fans. The analog control for the fans is 0x1400. The possible fan speed values for each shelf are:

- ATCA-6000: 0 to 10
- ATCA-6002: N/A. The fans always run at full speed and are not under the control of the Shelf Manager.
- ATCA-6006: 0 to 15
- ATCA-6014 and ATCA-6016: 0 to 15

You must set the control mode to manual to override the setting. Set the mode to automatic to return the fan speed to the Shelf Manager's control.

Tip: To change the minimum fan speed, see [Setting the minimum fan speed on page 45](#).

You can override the cooling algorithm:

- Using CLI commands. Enter:

```
platform-mgmt
controls
configure fan-speed
```

When you select a specific speed, the control mode is changed to manual. When you select the auto option, the fan is controlled by the Shelf Manager again.

- Using the HPI function `saHpiEventGet()` to receive temperature events automatically and the HPI fan control for each fan resource to control fan speeds. For details, see [Overriding the cooling algorithm with an HPI application on page 75](#).
- Using the SNMP variables:
`saHpiCtrlAnalogMode.0xffffffff.<fan-resourceID>.false.0x1400`
`saHpiCtrlAnalogState0xffffffff.<fan-resourceID>.false.0x1400`
from the HPI-B0101-MIB module.
 1. Set the analog control mode for the fan to manual(2).
 2. Set the analog control state for the fan to the new value representing the fan speed. Do not set the mode back to auto(1), unless you want to return to the Shelf Manager's cooling algorithm.
- Using the `hpiapp` example application interactively or as a programmatic example for writing your applications.

Overriding the cooling algorithm with an HPI application

The following steps illustrate how you can do this using HPI:

1. Open an HPI session using the active Shelf Manager IP address as DomainId (see [Writing HPI applications on page 57](#) for information).
2. Using the fan resources, set all fan controls to the SAHPI_CTRL_MODE_MANUAL mode, keeping their speeds unchanged. This places them in the system manager override mode. The fan resources can be identified by the associated entity type, SAPHI_ENT_COOLING_UNIT.
3. Subscribe for events on the session (see the events section of the *SAF HPI Specification*). Start listening for events in blocking mode by issuing the *saHpiEventGet()* function call with Timeout = SAHPI_TIMEOUT_BLOCK (see the function description in the *SAF HPI Specification*). This enables reception of events dynamically.
4. When a temperature assertion event of significant severity is received, feed it into the external cooling module. The cooling module can then use the fan resource's fan control to raise or lower the fan speeds.

For more information on HPI cooling and fan control, see the *SAF Mapping Specification*.

Changing shelf power properties

The shelf resource contains two controls that enable system managers to update shelf power properties in the Shelf Power Distribution record. In general, system managers should never need to change these properties. However, in some cases, a new board with high power requirements can exhaust the power budget of the shelf. System managers may need to adjust the power distribution so that more external current or a higher minimum voltage is available to the shelf.

WARNING! Use caution when changing these values, as the wrong power property values can lead to shelf and board damage. It is recommended that you check with Radisys Technical Support before making changes using these controls.

The controls are standard HPI analog controls and include:

- Control 0x1011: Max External Available Current
The minimum value is 5 amps and the maximum is 150 amps. This control reports the current value in tenths of amps. For example, a value of 10A will be reported as 100. This is how the value is stored in the shelf FRU record.
- Control 0x1012: Min Expected Operating Voltage
The minimum value is 36V and the maximum is 72V. This control reports the current value in 1/2 V increments. For example, a value of 40V will be reported as 80. This is how the value is stored in the shelf FRU record.

Saving and loading custom FRU configurations

HPI lets you restore a resource to the default factory settings, save and restore custom configuration settings, and apply a custom configuration for one FRU to another FRU of the same type.

You can manage FRU configurations:

- Using the standard HPI function *saHpiParmControl()*. For details, see [HPI parameter control on page 77](#).
- Using the SNMP variable *saHpiResourceParmControl.0xffffffff.<resourceID>.false* from the HPI-B0101-MIB module. Set the resource parameter control for the FRU to *restoreDefaultParm(1)* to restore factory defaults, to *saveParm(2)* to save the current settings, or to *restoreParm(3)* to restore the saved settings.
Note: The value you set remains while the parameter control operation is in progress. The undefined(0) value indicates completion and readiness to perform another operation.
- Using the *hpiapp* example application interactively or as a programmatic example for writing your applications.

See the sections that follow for more details and instructions for managing FRU configurations.

HPI parameter control

The shelf resources, along with all intelligent FRU resources and slot resources, support the parameter control feature. These resources have the SAHPI_CAPABILITY_CONFIGURATION bit set in the ResourceCapability mask of their resource presence table (RPT) entry.

There are three basic *saHpiParmControl()* actions:

- 0 – SAHPI_DEFAULT_PARM Loads the default settings for a resource and restores it to factory state. Use this option to reset the FRU associated with the resource to the factory default state.
- 1 – SAHPI_SAVE_PARM Saves the current configuration state of the resource and updates the configuration file. Use this option if the FRU associated with the resource needs to retain the current state following a power cycle, reactivation or re-insertion in the same slot.
- 2 – SAHPI_RESTORE_PARM Loads the configuration file saved by an earlier SAHPI_SAVE_PARM operation and restores the FRU associated with the resource to the custom state.

Note: If the saved configuration file is found during system startup or during FRU insertion (when the resource for the FRU is being created), the active HPI server will load it and initialize the resource to the custom state.

See the configuration section of the *SAF HPI Specification* for details on *saHpiParmControl()*.

Data saved as configuration parameters

The following data is stored as configuration parameters for all resources supporting the parameter control capability:

Table 7. Data stored as configuration parameters

All Resources	Each Sensor	Each Control
<ul style="list-style-type: none"> Resource ID Resource manufacturer ID Resource product ID Resource tag Resource severity Resource auto-extract timer (if configurable) 	<ul style="list-style-type: none"> Sensor number Thresholds and hysteresis (if supported) Enable state (if configurable) Event enable state (if configurable) Event assert and deassert masks (if configurable) 	<ul style="list-style-type: none"> Control number Control mode Control type Control state (if control is not ReadOnly)

HPI server parameter control configuration files

Each resource has two configuration files: one for default settings and one for custom saved settings.

The following are the file name formats:

%08X.hcf.deflt – Default configuration file name format

%08X.hcf.saved – Saved configuration file name format

%08X represents the ResourceId in 4-byte hexadecimal format.

For example, a resource with ID 0x70082 will have configuration files *00070082.hcf.deflt* and *00070082.hcf.saved*.

All configuration files are human-readable and may be updated using any standard text editor. However, Radisys recommends that manual modifications not be made to configuration files due to the possibility of human error. While default configuration files should not need to be updated, the preferable method of updating saved configuration files is to first make the necessary changes in the resource and then use the HPI call *saHpiParmControl()* to save the updated state of the resource.

How configuration files are stored

The default and saved configuration files are stored in the volatile RAM file system directory */tmp/shmgr/hpi* for faster read and write access. The HPI server loads the configuration files from this directory, which is created during Shelf Manager startup.

In addition to storing the files in the RAM file system, a persistent copy of all configuration files is stored in gzipped archive format in the non-volatile location */var/lib/shmgr* on the Shelf Manager module. During Shelf Manager startup when the HPI server daemon is getting initialized, the gzipped configuration archive is copied from */var/lib/shmgr* to the */tmp/shmgr/hpi* directory and unzipped. If the *HSD_LOAD_SAVED_CONFIGURATION* parameter is set to Y in the */etc/shmgr.conf* file, the HPI server subsequently loads all saved configuration files when it creates the associated resources.

Every time a new configuration is saved for a resource, its associated saved configuration file is updated in the RAM file system. The updated configuration file is also appended to the persistent configuration file archive in */var/lib/shmgr*. This preserves the configuration files for reuse when the Shelf Manager is rebooted.

How configuration files are created for a new resource

During HPI initialization (when resources are being created for the shelf, slots, and FRUs), initial default configuration files are automatically created if none are found in */tmp/shmgr/hpi*. If no corresponding saved configuration file is found for the resource in */var/lib/shmgr*, the initial default configuration file is copied as the *initial* saved configuration file (with the *.hcf.saved* suffix). You can later overwrite the initial saved configuration file in */var/lib/shmgr* to create a custom default configuration.

For your convenience, the Shelf Manager module's software image also comes with sample default configuration files for some of the known FRUs in the system, the slot resources and the shelf resource. These sample default configuration files reside in the */var/lib/shmgr* directory of the module's software image. Note that the sample default configuration file for slot resources, *SLOTS.hcf.deflt*, is copied as the default configuration file for all slot resources the first time a Shelf Manager module is initialized. The module does not have any configuration files prior to first initialization.

Creating a default configuration file for a new resource

If a new FRU is inserted in the shelf and no sample default configuration exists (for example, in the unlikely scenario that default configuration file creation failed during resource creation), the following procedure can be used to create one:

1. Insert the FRU in the shelf and check the HPI RPT for its resource. This can be done using an HPI 1.1-compliant user application.
2. Before any changes are made to the FRU, use the *saHpiParmControl()* function with:
 ResourceId = *ResourceId of the newly-created resource*
 Parm Action = SAHPI_SAVE_PARM.

If the parameter control operation succeeds, a new saved configuration file will have been created in */tmp/shmgr/hpi*. For example, for resource 0xB0090, a saved configuration file named *000B0090.hcf.saved* will be created.

3. Rename the *%08X.hcf.saved* file to *%08X.hcf.deflt*.

For example, for resource 0xB0090, the saved configuration file *000B0090.hcf.saved* should be renamed or moved to *000B0090.hcf.deflt*. This now becomes the default HPI configuration file for the new resource and will be loaded when the parameter control action SAHPI_DEFAULT_PARM is invoked on the resource by the HPI application.

Saving a configuration file to store custom settings for a resource

After a resource has been configured to a desired operational state, it is a good idea to save the configuration so it can be loaded whenever the FRU is re-introduced in the shelf in the same slot. If you need to restore the state of the resource later, you can load the saved configuration using the parameter control action SAHPI_RESTORE_PARM.

To create a saved configuration file for a resource, use the *saHpiParmControl()* function with:

```
ResourceId = ResourceId of the resource  
Parm Action = SAHPI_SAVE_PARM
```

If the parameter control operation succeeds, the saved configuration file for the resource will have been updated in */tmp/shmgr/hpi*.

Using a resource's configuration file for another resource of the same kind

To use a resource's saved configuration file for another resource of the same kind, rename the configuration file for the alternate resource. The example below illustrates this procedure as applied to two CPMs, but you can use the same procedure to create duplicate configuration files for any of the slot resources, the SCMs, the SPMs, or the node modules.

Example:

This example assumes you want to use the same configuration settings on two Compute Processing Modules (CPMs) in the shelf with the following ResourceId values:

Slot 4 – ResourceId 0x4008E

Slot 11 – ResourceId 0xB0090

1. Configure either one of the resources, in this case the CPM in slot 4 (ResourceId 0x4008E), to a desired state.
2. Save the configuration using the SAHPI_SAVE_PARM operation described in [Saving a configuration file to store custom settings for a resource](#) (above).

This creates the configuration file *0004008E.hcf.saved* in */tmp/shmgr/hpi* for the CPM resource 0x4008E. (0x0004008E is the slot 4 CPM's ResourceId in 4-byte hexadecimal format.)

3. Copy the *0004008E.hcf.saved* configuration file to *000B0090.hcf.saved* (0x000B0090 is the ResourceId for the slot 11 CPM in 4-byte hexadecimal format). This new file is now usable as the saved configuration file for the CPM resource 0xB0090 in slot 11.
4. The *saHpiParmControl()* function can now be used to load the saved configuration to the CPM in slot 11 using these values:

```
ResourceId = 0xB0090  
Parm Action = SAHPI_RESTORE_PARM
```


Error conditions

The parameter control operations SAHPI_DEFAULT_PARM or SAHPI_RESTORE_PARM will fail and return the error code SA_ERR_HPI_INVALID_DATA if a configuration file for one resource is loaded with another resource of a different kind.

This check is done by matching the Manufacturer ID and Product ID of the FRU. For example, if a CPM is replaced by a different type of module in the same slot, and a parameter control operation is invoked to load a configuration file saved for the original module, the operation will fail.

The parameter control operation will also fail if the necessary configuration file does not exist at all. In such cases, the error code SA_ERR_HPI_NOT_PRESENT will be returned.

Note: If a FRU's firmware is upgraded while the FRU is active in the shelf, the HPI implementation discards the current parameter control files for that resource (both *.deflt* and *.saved*) and creates new configuration files once the FRU is power cycled. Alternatively, the user can force the HPI implementation to update the parameter control files for the updated FRU by issuing the *saHpiDiscover()* function. You can also force a discovery using the CLI command *rediscoverShelf* or the SNMP object *saHpiDiscover.0* described in [HPI subagent initialization on page 139](#).

Managing Alarms and Events

This chapter describes:

- The Shelf Manager's role in managing alarms.
- How you can change the alarm reporting by assigning severities to sensor events.
- The Shelf Manager interactions with the alarm functions of the shelves.
- How you can override the Shelf Manager's default use of the shelf alarm indicators.

For tips on responding to specific alarms, see [Chapter 6, Troubleshooting, on page 98](#).

Shelf Manager role in alarm management

The Shelf Manager monitors conditions of modules and other shelf components and controls their operation to keep them working properly. It monitors the status of the shelf and corrects problems when necessary. The Shelf Manager reports events and anomalies to a system manager and responds to action requests from the system manager.

As part of this function, the Shelf Manager maintains the domain alarm table (DAT), a list of all active alarms. Alarms are added to the table when a condition asserts, and are removed when the condition de-asserts. The Shelf Manager also performs telco alarm management as defined in the *AdvancedTCA Base Specification*.

The hardware platform interface (HPI) provides a standard application programming interface (API) between the system manager and the Shelf Manager. This API allows the system manager to subscribe to sensor events, interact with the DAT, and read the domain event log.

Note: There is no physical alarm support on the ATCA-6002 shelf. Alarms are still added to the HPI domain alarm table but there are no physical alarm indicators (such as LEDs or audible sounds) available.

Domain alarm types and troubleshooting overview

Table 8 lists the alarm and sensor types along with sample descriptions and causes for the types of alarm maintained in the DAT. *<num>* is the sensor number (in hexadecimal) of the sensor generating the alarm.

Table 8. Alarm type descriptions

Alarm type	Sensor type	Alarm description	Cause
Threshold	Voltage	Sensor <i><num></i> Volt high	A measurement above an upper threshold was detected by a voltage sensor.
		Sensor <i><num></i> Volt low	A measurement below a lower threshold was detected by a voltage sensor.
	Current	Sensor <i><num></i> Cur high	A measurement above an upper threshold was detected by a current sensor.
		Sensor <i><num></i> Cur low	A measurement below a lower threshold was detected by a current sensor.
	Revolutions per minute	Sensor <i><num></i> Fan high	A measurement above an upper threshold was detected by a fan speed sensor.
		Sensor <i><num></i> Fan low	A measurement below a lower threshold was detected by a fan speed sensor.
	Temperature	Sensor <i><num></i> Temp high	A measurement above an upper threshold was detected by a temperature sensor.
		Sensor <i><num></i> Temp low	A measurement below a lower threshold was detected by a temperature sensor.
IPMB and communication	N/A (alarm generated by resource)	FRU Comm Lost	Shelf Manager communications to the FRU were lost over the IPMB; that is, the FRU is in the M7 (not present) state. See Responding to a communication lost alarm on page 111 .
	OEM (discrete)	Sensor <i><num></i> IPMB Dsbl	IPMB bus is disabled. See Responding to an IPMB disabled alarm on page 111 .
		Sensor <i><num></i> IPMB Iso	IPMB bus is isolated. See Responding to an IPMB isolated alarm on page 114 .
Shelf Manager status	OEM (discrete)	ShM Non Redundant	The active Shelf Manager has lost communication with its redundant peer. See Responding to a redundancy lost alarm on page 115 .
		Shelf FRU Data Invalid	The shelf FRU data was detected to be invalid. See Responding to a shelf FRU invalid alarm on page 116 .
FRU aggregate status	Operational	FRU Operation Impaired	FRU operation is disabled or degraded.
	Power unit	See values for threshold alarms above.	FRU power is out of range.
Resource utilization	N/A (user alarm added by process monitor, <i>rsys_pms</i>)	High CPU Utilization	CPU utilization for the Shelf Manager has exceeded 40% (reported by virtual sensor 0x1064).
		High Thread Utilization	Shelf Manager or HPI thread utilization exceeded the limit of 80% (reported by virtual sensors 0x1067 and 0x1068). See Table 18 on page 143 .
		High RAM Utilization	RAM utilization for the Shelf Manager has exceeded limit of 10 MB (reported by virtual sensor 0x1065).
User	N/A, added manually	User-defined	A user has added an alarm.

Each front module may generate one or more instances of these alarms depending on the managed sensors supported and the events enabled for each sensor. See the reference manual for each module for a list of managed sensors.

Threshold alarms

The first eight alarm types in [Table 8](#) are threshold alarms which indicate that a threshold sensor detected a voltage, current, temperature, or fan speed measurement that was out-of-range. Up to six default thresholds are specified for each threshold sensor:

- Upper Critical
- Upper Major
- Upper Minor
- Lower Minor
- Lower Major
- Lower Critical

Measurements between the Lower Minor and Upper Minor thresholds represent the normal operating range of the measurement. Measurements between the Upper Minor and Upper Major thresholds or between the Lower Minor and Lower Major thresholds represent minor alarm conditions. Measurements between the Upper Major and Upper Critical thresholds or between the Lower Major and Lower Critical thresholds represent major alarm conditions. And measurements above the Upper Critical threshold or below the Lower Critical threshold represent critical alarm conditions.

The Shelf Manager automatically adds an alarm to the DAT when the sensor generates an assert event for an alarm condition in which the measurement crossed an Upper threshold (going up) or crossed a Lower threshold (going down). The alarm severity indicates which threshold was crossed.

The alarm is automatically removed from the DAT when the sensor generates a deassert event for which the measurement crossed one (or more) thresholds going the other way and no longer represents an alarm condition. Hysteresis is used to prevent repeated alarm addition and removal when measurement variations occur close to one of the thresholds.

Table 9. Threshold alarm types

Alarm type	Description
Voltage alarms	Indicate that a power supply voltage threshold was crossed going up (high voltage) or going down (low voltage). Upper Minor and Lower Minor thresholds are typically within component requirements, and do not indicate an urgent condition. The Major and Critical events would indicate a power problem, and should be investigated.
Current alarms	Indicate that a power supply load current threshold was crossed going up (high current) or going down (low current). Current sensors are part of the two shelf power entry modules (PEMs). These events indicate that the 48V/60V load current has exceeded design levels, and the cause should be investigated.
Fan speed alarms	Indicate that a fan speed threshold was crossed going up (high speed) or going down (low speed). These events indicate that the fans are not rotating at the intended speed. In some cases this may be a temporary condition as the fans change from one target speed to the next. If the condition persists, contact Radisys Technical Support for help.
Temperature alarms	Indicate that a temperature threshold was crossed going up (high temp) or going down (low temp). Upper Minor temperature events are used by the Shelf Manager to control routine system cooling, and do not indicate an urgent condition.

Upper Major temperature events may indicate that the ambient temperature is too high, in which case you might need to lower the ambient temperature. They may also indicate that air flow into the shelf has become restricted, in which case you need to investigate the cause. Remove, clean, and reinsert the fan filter if it has become clogged.

Upper Minor and Upper Major temperature event assertions cause the fan speeds to increase in increments until the condition is cleared. A de-assertion event from the same sensor indicates that the condition is cleared, and the fan speed is decreased accordingly. A pattern of temperature event assertions and de-assertions may lead to an increase in the minimum fan speed. For more information, see [Adaptive fan floor algorithm on page 34](#).

When the Shelf Manager receives an Upper Critical temperature event, it attempts to power off the module associated with the sensor. Even while powered off, the sensors are still functional. The alarm condition may not be cleared immediately, if at all, so the fans may continue to run at a high speed.

IPMB and communication alarms

The IPMB and communication alarms indicate an abnormal condition for an IPMC IPMB interface or for communications between the Shelf Manager and a given FRU.

FRU communication lost alarms are generated by the resource instead of by a sensor. IPMB alarms are generated by OEM sensors. An OEM sensor has an 8-bit value and is defined by an OEM. OEM sensors are a subset of the discrete sensor type, which can have up to 16 bit-mapped states.

Shelf Manager status alarms

The Shelf Manager status alarms indicate that the Shelf Manager is not functioning normally or in full high availability (HA) mode.

Shelf Manager status alarms are generated by OEM sensors. An OEM sensor has an 8-bit value and is defined by an OEM. OEM sensors are a subset of the discrete sensor type, which can have up to 16 bit-mapped states.

FRU aggregate status alarms

The FRU aggregate status alarms indicate that a FRU has significant error conditions. These could include:

- Operational error conditions such as IPMC self-test failure or E-Keying errors.
- Power status errors based on threshold crossings.

Note: An additional temperature aggregate sensor exists on all FRU resources. Though this sensor does not generate separate alarms and events, you can read it for an overall view of the board's thermal status. Thermal conditions of a FRU will generate alarms from associated temperature sensors.

For details on the possible states of the FRU aggregate status sensors, see [Table 19 on page 144](#).

Resource utilization alarms

The resource utilization alarms indicate that the Shelf Manager is using a high percentage of the host module's resources. If there is another process on the module using all of the resources, the sensor does not generate an alarm.

A process monitoring service, *rsys_pms*, runs as a background process on the module and every 30 seconds checks the utilization percentage for the CPU, RAM, and Shelf Manager and HPI threads. If the pre-defined thresholds are crossed in two consecutive runs, *rsys_pms* generates a user alarm for the overutilized resource.

User alarms

Users can add and remove user alarms via HPI to report user-defined alarm conditions. You can specify the maximum number of user alarms allowed as described in [Configuring HPI alarms on page 46](#).

Domain alarm table entries

The alarm strings are based on the associated sensor events. Text within the alarm string indicates the FRU location, sensor number, and severity of the sensor event. The reference manuals for each module contain detailed information on their associated sensors.

The domain alarm table (DAT) can hold a maximum of 1024 alarm entries. The alarm entries include system alarms, which are generated automatically, and any user-defined alarms.

System alarms will display a description of ACTIVE (*) when in an unacknowledged state. The (*) indicates that the alarm has not yet been acknowledged. User-defined alarms, however, will show a description of ACTIVE () even in an unacknowledged state, because it is assumed the user is aware of the condition that they defined.

Assigning event severities

The events generated by sensors are assigned severities automatically, but you can override the default settings by assigning new severities to specific events generated by individual sensors. You can assign severities to discrete, digital, and OEM sensors (with the exception of IPMB sensors).

To assign severities, add entries to the Shelf Manager alarm configuration file (*/etc/shmgralarm.conf*) on the Shelf Manager module or the SCM hosting the active Shelf Manager. Each entry applies to a single sensor, although the sensor may apply to multiple of the same type of FRU in the shelf. For example, a change to the severity of a fan sensor event would apply to all of the fans in the shelf if the fans are identical and are running the same firmware version.

As an example of a severity change, if the default severity of major for the PEM power fault event is not suitable, it can be changed to critical, minor, or informational.

Changing the severity affects:

- The event entry in the domain event log and the shelf's system event log.
- The presence of an alarm and the alarm severity in the domain alarm table. Critical, major, and minor events are included in the DAT as alarms. Informational and OK events are not included in the DAT.
- Any audible alarms, LEDs, or displays that indicate the presence of alarms for the platform.

Collecting information for the alarm configuration file

For each type of FRU and each sensor for which you will assign severities, you must provide identifying information. [Table 10](#) explains how to obtain the information to identify the FRU type.

Table 10. Information needed to identify a FRU type

FRU information needed	Method for obtaining it	Notes
Manufacturer ID	From the module's Linux prompt, enter: rsys-ipmitool -t <FRU_IPMB_addr> mc info	The manufacturer ID and the firmware revision are shown in decimal, and the product ID is shown in both decimal and hex. Either the decimal or hex value can be used in the configuration file. A prefix of 0x must precede any hex value.
Product ID		
Firmware revision ^a		
Site type	See Resource types of slots and FRUs in the RPT on page 135 .	For non-intelligent FRUs, use the site type of the proxy FRU that represents it.
^a The new event severity is assigned only to FRUs with the listed firmware version or a later version. For example, if the firmware version is listed as 3.0, the event severity would be assigned to any FRU with a firmware version of 3.0 or later, such as 3.1, 3.2, and 3.2.4.3.		

Table 11 lists the information needed to identify the sensor, the event state, and the new severity to assign.

Table 11. Information needed to identify a sensor and event state

Sensor information needed	Method for obtaining it	Notes
Sensor index	<p>Retrieve a list of the FRU's sensors from the RPT using an HPI application.</p> <p>For example, from the module's Linux prompt, use the <i>hpiapp</i> example application as follows:</p> <ol style="list-style-type: none"> 1. Start the application by entering: hpiapp 2. Identify the FRU's resource ID by entering: 3 0 3. Request a sensor list by entering: q s b 4. When prompted, enter the resource ID. 	<p>The sensor index is based on the order of the sensors in the resource presence table. To identify the sensor index, assign numbers to the list of sensors, starting with 0 (zero) at the top of the list.</p>
	<p>Alternatively, retrieve the sensor list using the CLI:</p> <ol style="list-style-type: none"> 1. Access the platform-management CLI as described in Accessing the platform-management CLI on page 56. 2. Specify the applicable FRU type. For example, for a front module, enter: front <number> 3. Request a sensor list. Enter: show sensors all 	
Event state	<p>See the reference manual for the FRU to locate the numeric event state values for the sensor. For shelf-specific FRUs such as PEMs and fans, see the reference manual for the platform.</p>	<p>For non-intelligent FRUs associated with the shelf, look for their sensors in the manual that covers the proxy (for example, refer to the appropriate <i>Shelf Peripheral Module Hardware Reference</i> for the ATCA-6000, which uses one or two SPMs as proxy FRUs).</p>
Severity to assign	<p>Select one of the severities defined by HPI:</p> <ul style="list-style-type: none"> 0 = Critical 1 = Major 2 = Minor 3 = Information 4 = OK 	<p>Critical, major, and minor events are considered alarms and are added to the DAT.</p>

Configuration file structure and contents

The `/etc/shmgralarm.conf` file uses this structure for its contents:

FRU identification

Sensor identification, state-to-severity assignment, state-to-severity assignment

Sensor identification, state-to-severity assignment

FRU identification

Sensor identification, state-to-severity assignment

Sensor identification, state-to-severity assignment

Sensor identification, state-to-severity assignment, state-to-severity assignment

Example severity configuration

The configuration syntax is defined and exemplified in the default configuration file. [Table 12](#) shows an example of information collected for a FRU and two of its sensors. The corresponding configuration file contents are shown below the table.

Table 12. Example FRU and sensor information

FRU information		Sensor information			
Manufacturer ID	0x010F1	Sensor index	0x7	0x16	
Product ID	0x0B98	Event state	0002	0001	0002
Firmware revision	5.86	Severity to assign	1	3	1
Site type	0x00				

Example configuration file contents:

```
fru=SCM,0x010F1,0x0B98,5.86,0x00
sensor=0x7, 0002:1 # -48V Fuse Fault
sensor=0x16, 0001:3, 0002:1 # PIM Alarm
```

Rules for configuration file contents

- The FRU must be assigned a user-defined FRU name ("SCM" in the example above). For readability, it is recommended that you assign a name associated with the FRU's actual name; however, you can use anything as long as the name is unique within the file.
- The sensor index must be unique within the section for the FRU. Therefore, multiple severity assignments for the same sensor must appear in the same entry. (See the multiple assignments for sensor 0x16 in the example above.)
- Any hex number must be preceded by "0x".
- Blank lines are ignored.
 - A pound (#) sign indicates a comment that continues to the end of the line.
 - The comma (,) and colon (:) characters are reserved for use in separating fields as shown above. They must not be used in user-defined FRU names.

Alarm configuration procedure

Changes to the alarm configuration file take effect when the Shelf Manager is restarted. This procedure explains how to make the change take effect when the shelf has redundant SCMs.

To change the event severities:

1. Modify the `/etc/shmgralarm.conf` file on the SCM hosting the active Shelf Manager.
Tip: To check the role of an SCM, you can use this command from the SCM Linux prompt:
`shmgr role`
2. Stop the Shelf Manager on both the active and standby SCMs. Use this command from the Linux prompt of each SCM:
`shmgr stop`
3. Start the Shelf Manager on the SCM that has the updated configuration file, by entering:
`shmgr start`
4. Start the Shelf Manager on the other SCM:
`shmgr start`

The configuration file from the active SCM is updated on the standby SCM when the standby is started.

Tip: If you need to restore the default configuration, perform the configuration procedure above, replacing the `/etc/shmgralarm.conf` alarm configuration file with this file containing the defaults:

`/usr/share/shmgr/shmgralarm.defs`

ATCA-6000 shelf display and alarm indicators

The shelf display panel (SDP) on the ATCA-6000 shelf consists of the following:

- LCD display
- Shelf Power LED
- Alarm LEDs to indicate Minor, Major, and Critical alarms
- Audible alarm with an Alarm Acknowledge button

The LCD display on the SDP supports two scrolling 40-character lines to display the eight most recent alarms in the DAT. [Figure 9 on page 93](#) shows the alarm format used on the LCD display. Three separate alarm LEDs on the SDP indicate the presence of critical, major, and minor alarms in the DAT. An audible alarm on the SDP indicates the highest severity alarm. Telco alarm contacts, active on only the active SPM, indicate critical, major, and minor alarms.

Unless the automatic control has been overridden by the system manager, the Shelf Manager automatically controls the display and alarm indicators as follows:

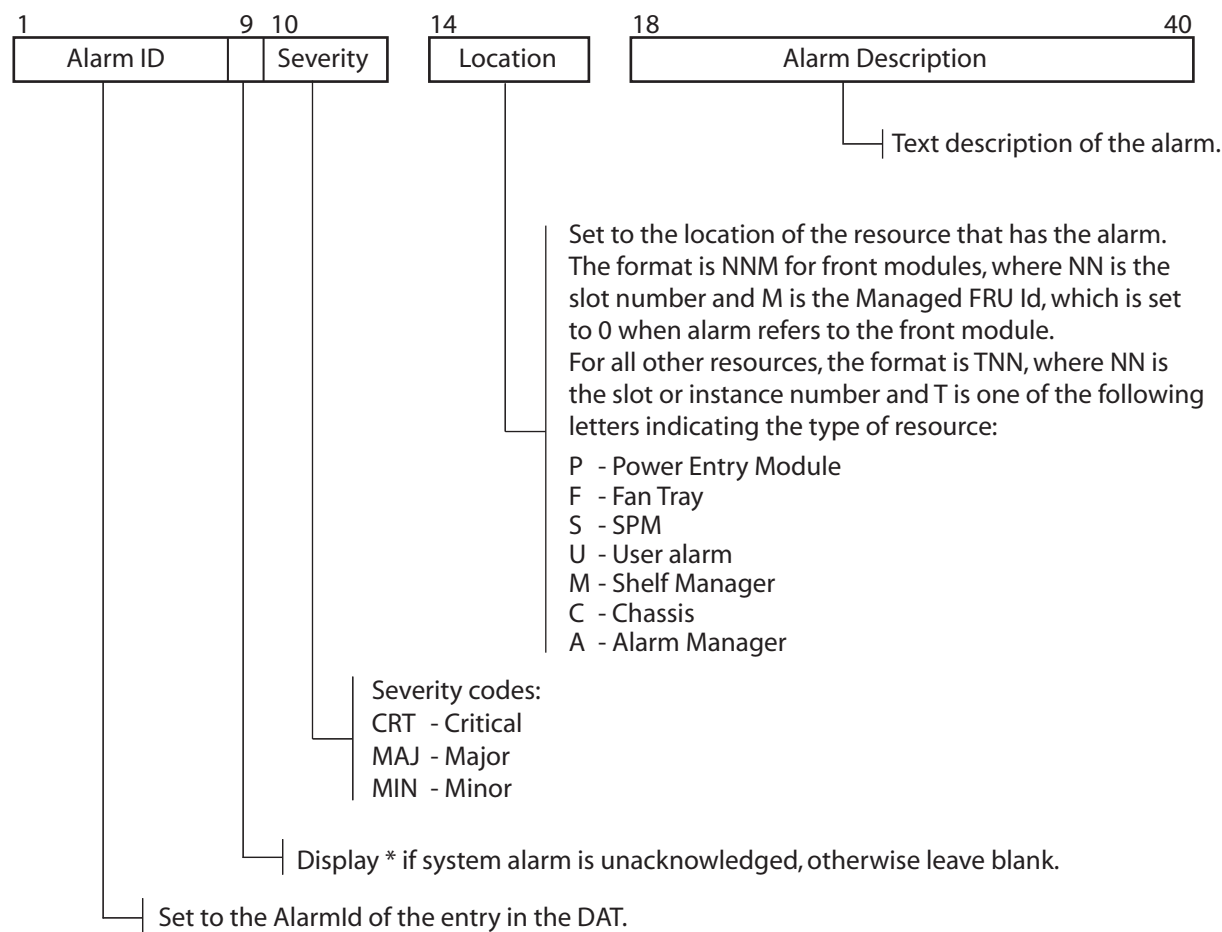
- Turns on the Critical, Major, or Minor LED on the SDP when it detects a critical, major, or minor alarm.
- Turns off the Critical, Major, or Minor LED on the SDP when all critical, major, or minor alarms are removed.
- Sounds an alarm when unacknowledged alarms exist in its domain alarm table.
- Stops sounding the alarm when all alarms have been acknowledged.
- Acknowledges all alarms in its domain alarm table when the Alarm Acknowledge button is pressed and released twice within 3 seconds.
- Sends an event to the system manager interface (HPI) when the Alarm Acknowledge button is pressed and released twice within 3 seconds.

If the Alarm Acknowledge button is pressed for more than 3 seconds, the Shelf Manager gradually reduces the contrast on the LCD display until the button is released. If the lowest contrast setting is reached and the button has not been released, the Shelf Manager sets the contrast to its highest level before continuing to reduce it. The cycle continues until the button is released.

- Displays the message “System Normal” on the SDP if there are no Alarms in its domain alarm table.
- Displays the last 8 Alarms in the domain alarm table on the SDP. The format of the messages is shown in [Figure 9](#). The messages are scrolled up at a rate of 1 second per row.

The telco alarm outputs are controlled by the Shelf Manager, but provide the capability for the system manager to override the state of the alarm contacts.

Figure 9. Format of alarm messages on the SDP



ATCA-6006 shelf alarm indicators

The optional shelf alarm panel on the ATCA-6006 shelf consists of the following:

- Shelf Power LED
- Alarm LEDs to indicate minor, major, and critical alarms
- Audible alarm
- Audible alarm reset pinhole
- Alarm acknowledge pinhole
- Telco alarm connector

The telco alarm outputs and shelf alarm indicators are controlled by the Shelf Manager through HPI, but provide the capability for the system manager to override the state of the alarm outputs. Unless the automatic control has been overridden by the system manager, the Shelf Manager automatically controls the display and alarm indicators as follows:

- Turns on the Critical, Major, or Minor LED on the alarm panel when it detects a critical, major or minor alarm.
- Turns off the Critical, Major, or Minor LED on the alarm panel when all critical major or minor alarms are removed.
- Restores the state of the Alarm LEDs based on alarms currently in the DAT.
- Sounds an alarm when unacknowledged alarms exist in its domain alarm table.
- Stops sounding the alarm when all alarms have been acknowledged.
- Acknowledges all alarms in its domain alarm table when the Alarm Acknowledge button is pressed and released.
- Sends an event to the system manager interface (HPI) when the Alarm Acknowledge button is pressed and released.

When the audible alarm reset is pressed (using a pointed object such as a pen), the following occurs:

- The audible alarm is silenced temporarily.
- The LEDs blink and the corresponding alarm relays are deactivated.
- The audible alarm reset state ends after 20 seconds, and the telco alarm returns to its previous state.

ATCA-6014 and ATCA-6016 shelf alarm indicators

The ATCA-6014 and the ATCA-6016 shelves both have a shelf alarm panel (SAP) and a shelf alarm display (SAD). The SAP is a telco alarm connector located on the front of the ATCA-6014 shelf and on the rear of the ATCA-6016 shelf. The SAD consists of alarm LEDs that indicate minor, major, and critical alarms, and is located on the front of both the ATCA-6014 and the ATCA-6016.

The alarm LEDs are controlled directly by the telco alarm states and the telco alarm outputs are controlled by the Shelf Manager through HPI.

The Shelf Manager automatically controls the display and the alarm indicators as follows:

- Turns on the Critical, Major, or Minor LED on the shelf alarm display when it detects a critical, major or minor alarm.
- Turns off the Critical, Major, or Minor LED on the shelf alarm display when all critical major or minor alarms are removed.
- Restores the state of the alarm LEDs based on alarms currently in the DAT.

Managing shelf alarm indicators

This section gives methods for overriding the Shelf Manager's default responses to shelf alarm indicators on the ATCA-6000 and the ATCA-6006 shelves. For more information, see the *SAF HPI Specification* and the *SAF Mapping Specification*.

Note: The shelf alarm indicators on the ATCA-6002, the ATCA-6014, and the ATCA-6016 shelves cannot be overridden.

Accessing the alarm indicator HPI controls

Access to the HPI controls for the alarm indicators is through a proxy FRU or directly to an intelligent shelf alarm panel FRU, depending on the shelf. [Table 13](#) lists the resource ID of each of these FRUs. For the IPMB address of each FRU, see [Table 17 on page 134](#).

Table 13. HPI resource IDs of FRUs for accessing the alarm indicator HPI controls

Shelf	FRU that gives access to alarm controls	HPI resource ID
ATCA-6000	Virtual SPM	0x1C250
ATCA-6006	Shelf alarm panel (an intelligent FRU)	0x10652

To override the Shelf Manager's automatic control of the alarm indicators, the HPI user must change the control mode from automatic to manual, and set the control state to the desired value for the given control. These are both done using the *saHpiControlSet()* function, documented in the *SAF HPI Specification*.

- The CtrlMode must be changed from SAHPI_CTRL_MODE_AUTO to SAHPI_CTRL_MODE_MANUAL.
- The CtrlState must be set to the desired value, using the appropriate union for the control type.

The controls for the telco alarm contacts (Alarm Critical, Alarm Major, and Alarm Minor) are standard HPI controls. On the ATCA-6000 shelf, the controls for the audible alarm and LCD panel are also standard HPI controls.

The controls for the Alarm LEDs on the ATCA-6000 shelf are OEM controls, and are defined in detail in the LEDs section of the *HPI-to-AdvancedTCA Mapping Specification*. You can set any desired blinking pattern for these LEDs, but by default only the on and the off states are used by the Shelf Manager for alarm indication purposes. The supported colors are listed below, and they may be set to on, off, or blinking. Middleware vendors may provide a simpler interface to the user to update HPI LED controls.

Table 14. HPI controls for alarm indicators

Control #	Name	Type	Notes	Applies to
3	Critical Alarm LED	OEM	The supported color is Red.	ATCA-6000
4	Major Alarm LED	OEM	The supported colors are Red (default) and Amber.	ATCA-6000
5	Minor Alarm LED	OEM	The supported color is Amber.	ATCA-6000
7	Audible	Stream	The two-byte stream represents a two-second duty cycle, with each nibble representing one-half second of beeping. The user can select any stream pattern to achieve the desired beeping pattern. By default, the Shelf Manager uses the following patterns: <ul style="list-style-type: none"> • 0xF0, 0xF0 for critical alarms (a short beep each second) • 0xFF, 0x00 for major alarms (a long beep every 2 seconds) • 0xF0, 0x00 for minor alarms (a short beep every 2 seconds) 	ATCA-6000
8	LCD Panel	Text	The two-line stream of 40 bytes each can be set to any desired set ASCII characters to provide a fixed message or a blank message.	ATCA-6000
9	Alarm Critical	Digital	Use SAHPI_CTRL_STATE_ON to close the critical alarm contact. Use SAHPI_CTRL_STATE_OFF to open the critical alarm contact.	ATCA-6000, ATCA-6006
10	Alarm Major	Digital	Use SAHPI_CTRL_STATE_ON to close the major alarm contact. Use SAHPI_CTRL_STATE_OFF to open the major alarm contact.	ATCA-6000, ATCA-6006
11	Alarm Minor	Digital	Use SAHPI_CTRL_STATE_ON to close the minor alarm contact. Use SAHPI_CTRL_STATE_OFF to open the minor alarm contact.	ATCA-6000, ATCA-6006

Parameter values for setting alarm controls

The following list identifies the possible *saHpiControlSet()* function parameter values:

- SessionId = *<as returned by the saHpiSessionOpen() function>*
- ResourceId = *<resourceId of the FRU giving access to alarm indicators, see Table 13 on page 95>*
- CtrlNum = *<control number of the given control, see Table 14 on page 96>*
- CtrlMode = SAHPI_CTRL_MODE_MANUAL
- CtrlState = *<one of the following>*

For Alarm LED controls: (ATCA-6000 only)	<ul style="list-style-type: none"> • .Type = SAHPI_CTRL_TYPE_OEM • .StateUnion.Oem.MId = 0 • .StateUnion.Oem.BodyLength = 6 • .StateUnion.Oem.Body = <i><refer to section 4.5.7 of the HPI-to-AdvancedTCA Mapping specification></i>
For Audible control: (ATCA-6000 only):	<ul style="list-style-type: none"> • .Type = SAHPI_CTRL_TYPE_STREAM • .StateUnion.Stream.Repeat = SAHPI_TRUE • .StateUnion.Stream.StreamLength = 2 • .StateUnion.Stream.Stream = <i><two-byte pattern to represent desired beeping pattern></i>
For LCD Panel control: (ATCA-6000 only)	<ul style="list-style-type: none"> • .Type = SAHPI_CTRL_TYPE_TEXT • .StateUnion.Text.Line = 1, 2, or SAHPI_CLN_ALL_LINES • .StateUnion.Text.Text.DataType = SAHPI_TL_TYPE_TEXT • .StateUnion.Text.Text.Language = SAHPI_LANG_ENGLISH • .StateUnion.Text.Text.DataLength = <i><0-40></i> or <i><0-80></i> if Line is SAHPI_TLN_ALL_LINES • .StateUnion.Text.Text.Data = <i><user-specified ASCII text></i>
For Telco Alarm contact controls: (ATCA-6000, ATCA-6006):	<ul style="list-style-type: none"> • .Type = SAHPI_CTRL_TYPE_DIGITAL • .StateUnion.Digital = SAHPI_CTRL_STATE_OFF or SAHPI_CTRL_STATE_ON

Troubleshooting

This chapter provides detailed information on the following issues:

- Available diagnostic tools for FRU and Shelf Manager problems.
- General troubleshooting procedures that apply to alarms, system events, and sensor and temperature values.
- IPMI and IPMB topics and troubleshooting procedures.
- Issues related to FRU or shelf operation, such as responding to temperature and other sensor alarms or dealing with loss of communication with a FRU.
- Issues related to the Shelf Manager itself and its operation.
- Issues related to shelf FRU information, including how to identify shelf FRU information problems and how to save and restore shelf FRU information using the `frurw` command.

Important: Many of the examples in this chapter show the command syntax for the Radisys CLI. For details on syntax and options, see the *CLI Reference*.

Diagnostic tools

To diagnose both FRU and Shelf Manager problems:

- Use the platform-management CLI, which is intended for use by field personnel with limited knowledge of IPMI and ATCA.
- Use IPMI diagnostic tools. For details, see [IPMI and IPMB troubleshooting topics on page 104](#).
- Use the `shmgrdiag` Linux utility to get the status of the Shelf Manager and the Shelf Manager module and to gather files that can be sent to Radisys when contacting Technical Support. For details, see [Using the Shelf Manager diagnostics utility](#).

Note: This utility is supported on the ATCA-6002 shelf when the Shelf Manager is running in enhanced mode; however, the utility does not look for redundancy and does not check for LAN synchronization interface status.

- Use the sample HPI application, `hpiapp`. This application presents the status of the shelf and FRUs from an HPI perspective. The sample application is intended for developers familiar with the HPI specification.
- Use third-party HPI tools, such as OpenHPI.

Using the Shelf Manager diagnostics utility

The *shmgrdiag* utility gathers information on the current state of the Shelf Manager, the Shelf Manager module resource usage, and the key IP interfaces.

To run the utility from the Shelf Manager module's Linux prompt, enter:

```
shmgrdiag
```

The diagnostics display as they run and list the types of diagnostics being run as well as a summary of the results. Files containing the results are placed in the */var/lib/shmgr/diagnostics* directory and are saved permanently (bind mounted as described in the *Software Guide*). The files are especially helpful to Radisys Technical Support if you contact them for assistance.

The files include the Shelf Manager log files, the software package information, the top 10 processes running at the time, and the kernel buffer messages (*dmesg.log*).

Note: An equivalent CLI command, *diagnose*, is available in the platform-management CLI from shelf-management mode. The CLI command is available when the Shelf Manager is running. The *shmgrdiag* Linux utility can be used even when the Shelf Manager is not running.

General troubleshooting procedures

This section explains how to acknowledge and view alarms, view system events, and check sensor and temperature values.

Responding to alarms

You can use platform-management CLI commands to perform common procedures in response to alarms. For troubleshooting information on specific alarms, see [Responding to specific alarms on page 111](#).

To acknowledge alarms and silence the audible alarm

You can do one of the following to acknowledge alarms and silence the audible alarm:

- Use one of these alarms mode commands.
 1. `acknowledge all`
 2. `acknowledge alarmld`
 3. `acknowledge severity`

Notes:

- The `acknowledge alarmld` and `acknowledge severity` commands do not silence the audible alarm when alarms with other IDs and severities are present.
- The ATCA-6014 and ATCA-6016 shelves have no audible alarm.
- On an ATCA-6000 shelf, push the acknowledge button on the shelf display panel twice within three seconds. On an ATCA-6006 shelf, push the alarm acknowledge button on the shelf alarm panel, or push the audible alarm reset button to temporarily silence the alarm without acknowledging it.

To view a summary of current alarms

To view a summary of current alarms, including the number of alarms by severity, use the command `show alarmSummary`.

To view alarm details

To view details of alarms, use the alarms mode command `show all`.

To access alarms using alternative methods

You can use one of the following alternative methods to access alarms.

- You can access alarm information through *hpiapp* as follows:


```
hpiapp
d
0
```
- The shelf display panel (SDP) on the ATCA-6000 shelf cycles through a list of the current alarms.

Viewing the system event log

You can view a module's local SEL or the shelf's SEL using the open-source *ipmitool* utility or the *rsys-ipmitool* supplied by Radisys. You have these alternatives:

- For a shorter listing that contains fewer details:
 - a. Log in to the module and access the CLI. To view the shelf's SEL, log in to the active SCM and access the CLI.
 - b. Enter:


```
blade-mgmt
show sel
```
- For a longer list that identifies the module that generated each event and contains details about the system progress checkpoint events:
 - a. Log on to the Shelf Manager module (or active SCM) and access the Linux shell.
 - b. Enter:


```
rsys-ipmitool -t <IPMB> sel list -v
```

<IPMB> is the IPMB address of the module containing the local SEL you want to view. To view the shelf's SEL, specify an IPMB address of 0x20.

For the type of information shown in the local and shelf SELs, see [System event log \(SEL\) on page 22](#).

Viewing the domain event log

You can view the domain event log for a single FRU using the platform-management CLI.

Syntax: <slot_type> <slot_num>
 show events all

Example: front 3
 show events all

You can view the domain event log for the shelf through *hpiapp* as follows:

```
hpiapp
e
1
1
```

For the type of information shown in the domain event log, see [Domain event log \(DEL\) on page 23](#).

Investigating sensor values

To view the sensor value for a sensor that has generated an alarm:

1. Write down the FRU slot identified by the alarm and the severity of the alarm.
2. In the platform-management CLI, to access the front mode of an identified slot enter:

Syntax: `front <slot_num>`

3. Display the event log for the FRU using the command `show` with the *events* option.

Syntax: `show { events all | informational | minor | major | critical }`

4. Locate the event that corresponds to the alarm. The event includes the sensor name and value (state).

Syntax: `show events { all | informational | minor | major | critical }`

To view the value of a known sensor.

1. In the platform-management CLI, access the slot mode of the FRU containing the sensor.
2. Display the sensor using the command `show` with the *sensors* option.

Syntax: `show { sensors { all | sensorId <id> } }`

Note: To investigate sensors that apply to the Shelf Manager, use the shelf-management mode command `show sensors`.

Viewing temperature readings

To view temperature readings for a FRU

1. In the platform-management CLI, access the front mode of the FRU containing the sensor.
2. Display the sensor using the `show` command with the *sensors* option.

Syntax: `show { sensors { all | sensorId <id> } }`

To view approximate temperature readings for the shelves

On the ATCA-6000 shelf:

Each SPM has two sensors to measure the intake temperature, and each fan has an exhaust temperature sensor. Check the intake temperatures and compare them to the exhaust temperatures, as follows:

1. Use *hpiapp* to query the resource ID of an SPM:

```
hpiapp
3
0
```

2. List the sensors for the SPM resource:

```
q
s
b
<resourceId>
```

3. Note the intake temperatures.
4. Repeat steps 1, 2, and 3 for the other SPM.
5. In the platform-management CLI, access the fan slot mode of one of the fans.
6. Display the sensor using the command `show` with the *sensors* option.

Syntax: `show { sensors { all | sensorId <id> } }`

7. Repeat steps 5 and 6 for the other fans.
8. Compare the intake temperatures to the exhaust temperatures.

If the exhaust air temperature increases while the intake air temperature stays stable, this indicates that the temperature inside the shelf is increasing.

Be sure that the intake temperature remains within the environmental specifications for the shelf. See the *Platform Hardware Reference* for specifications.

Note: For help in identifying the physical locations of the numbered PEMs and fans, see the descriptions in [IPMB addresses of slots and FRUs on page 134](#).

On the ATCA-6002 shelf:

Air enters through the fan tray on the right side (when viewed from front) and exits through the left fan tray. Use the temperature sensors on the front modules and RTMs to get an approximate reading of the inlet and outlet temperatures.

On the ATCA-6006 shelf:

Air enters through the fan tray on the right side (when viewed from front) and exits through the left fan tray. Check the right fan temperature (slot 1) and compare it to the left fan (slot 2), as follows:

1. In the platform-management CLI, access the fan slot mode of one of the fans.
2. Display the sensor using the command `show` with the *sensors* option.
3. Repeat steps 1 and 2 for the other fan tray.
4. Compare the temperatures between the fan trays.

If the exiting air temperature increases while the intake air temperature stays stable, this indicates that the temperature inside the shelf is increasing.

Be sure that the intake temperature remains within the environmental specifications for the shelf. See the *Platform Hardware Reference* for specifications.

On the ATCA-6014 and ATCA-6016 shelves:

Each fan tray has an exhaust temperature sensor. Check the exhaust temperatures as follows:

1. In the platform-management CLI, access the fan mode for one of the fans.
2. Display the exhaust sensor using the command `show` with the *sensors* option.
3. Repeat steps 1 and 2 for the other fan trays.
4. Average the exhaust temperatures to get an aggregate exhaust reading.

IPMI and IPMB troubleshooting topics

This section provides information on IPMI and IPMB topics such as specification compliance, capacity and performance, message structure and flow, bus operation, bus redundancy, and possible failure symptoms.

In order to diagnose IPMB and IPMC problems, the following tools are available:

- The IPMC console is available on all Radisys IPMCs.
- IPMB analyzers are available that can show the message transactions on the bus in real time.
- For IPMB problems each blade generates IPMB events that contain the error cause in them along with the IPMB that is disabled.

IPMI specification compliance

The Radisys platforms are fully compliant with IPMI V1.5. Some features of IPMI v2.0 are supported by Radisys blades, such as the RMCP+ protocol used for Serial Over LAN on the CPMs and the version change sensor used to report IPMI FW updates. There are no plans at this time to fully comply with IPMI V2.0, however.

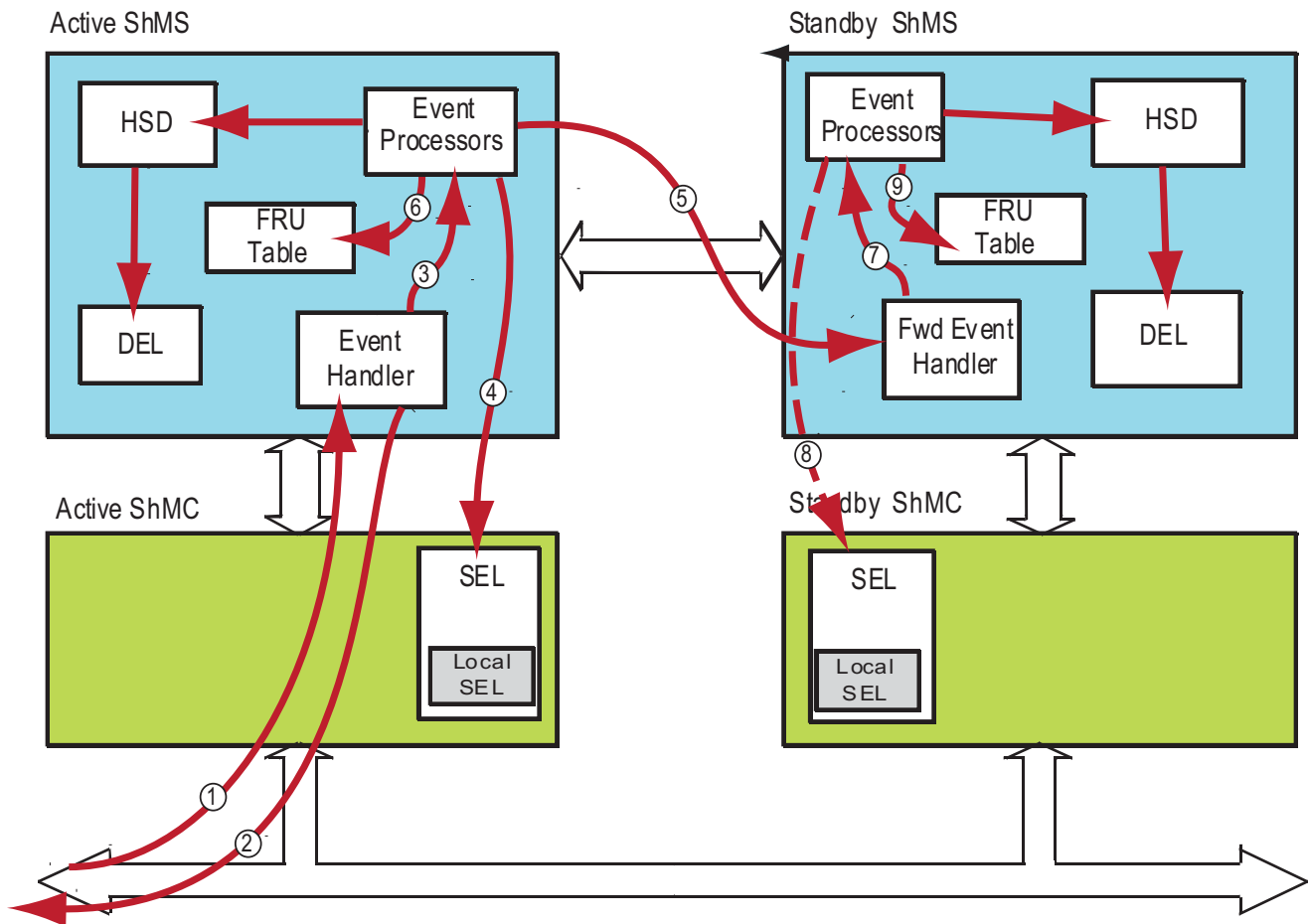
IPMB capacity and performance

The Radisys platform supports standard mode (100kbs) I²C communication in accordance with the PICMG specification. Fast mode (400Kbs) I²C communication is not supported at this time.

IPMI frame structure and message flow

There are no frames in IPMB messages. [Figure 10](#) shows the message flow through an SCM pair for IPMB messages.

Figure 10. IPMI Message Flow



IPMB bus operation

Two IPMB buses (IPMB-A and IPMB-B) provide the primary means of communication between ShMCs. The active ShMC on the SCM is also the IPMC for the SCM FRU, the COM-E, and the SCM RTM. The active ShMC performs the same operations other IPMCs do to maintain their FRUs. Messages sent to address 0x82/0x84 LUN 0 are all processed by the IPMC portion of the ShMC and are not forwarded to the ShMS. In addition, the ShMC performs the tasks of arbitration (determining active/standby state) and forwarding messages over the serial interface to the ShMS. The ShMC is basically a superset of the IPMC.

Only the switching portion of the SCMs act in Active-Active mode. The ShMCs are never in dual active state unless there's a catastrophic bus fault that does not allow any communication between the SCMs. The ShMCs arbitrate to become active or standby as soon as they initialize. If they ever become dual-active, they are constantly probing each other's states and would quickly notice the state mismatch and do re-arbitration.

The IPMB buses A and B are redundant and no synchronization occurs between the two buses. The ShMCs that are connected to multiple IPMBs follow a round-robin arbitration protocol. Requests are transmitted in round-robin manner on the two IPMBs.

If an ShMC fails to receive a response message from any slot (for example slot#3), it retransmits the request message up to three times at 250 msec intervals. The 250 msec interval is suggested by PICMG to allow for managed FRU communication, thus it can take up to 750 msecs to time out.

If a client cannot access IPMC right away (due to BUS busy), the IPMC keeps trying to arbitrate on the bus until it gets access and can transmit the bytes. Other functions are not pending in this state. A messaging queue is maintained and messages that need to get transmitted are put in this queue. The low-level I²C driver picks them up from this queue for transmission. All other IPMC functions carry on while this is happening. The tasks of hot-swap, sensor, FRU data, etc. are all handled in a separate thread.

IPMB bus redundancy

The same message is not transmitted simultaneously on the same bus. Messages are transmitted in a round-robin manner. So if message 1 is transmitted on bus A, message 2 will be on bus B, message 3 will be on bus A and so on. The IPMC keeps track of what bus was used for the last transaction and uses the other bus for the next. If a bus is isolated or an error is detected (reported by the IPMB-0 sensor) then all transactions happen on the other available bus.

IPMB Isolation causes are noted in the PICMG spec itself where it defines the IPMB-0 sensor event states. The most common causes are as follow:

1h = Unable to drive clock HI

2h = Unable to drive data HI

3h = Unable to drive clock LO

4h = Unable to drive data LO

5h = Clock low timeout

6h = Under test (the IPM Controller is attempting to determine if it is causing a bus hang)

7h = Undiagnosed Communications Failure

Each IPMC that is not the active ShMC monitors its “own side” of the bus. In other words, each IPMC is monitoring its own I²C interfaces that are connected to IPMB-A or B. Periodically it checks for errors on the I²C interfaces using registers in the IPMC. If an error is detected, the IPMC isolates itself from the IPMB, resets itself, and tries to rejoin the bus. If the IPMC is not successful, it generates the IPMB-0 sensor event via the other bus and keeps itself isolated from the error-prone bus. The IPMC periodically (every 250 msec) tries to reset the I²C interface and rejoin the error-prone bus until the error corrects.

The active ShMC has the responsibility of monitoring not only its own side of the IPMB but also the external IPMB. It can thus detect if some other error-prone IPMC in the chassis is holding one of the IPMBs down, and isolate itself from that bus. All transactions are then done on the other bus. Periodically the ShMC will join the error-prone bus to test if the error is still there.

IPMB and IPMC failure problems

All IPMCs arbitrate to get access to the bus, but messages can be missed if there is a lot of traffic on the bus (e.g. running a stress test), and all messaging buffers on the IPMC are full. Each IPMC retries requests three times and the ShMS retries each request 5 times. Also, hot-swap events are guaranteed to be received because they get retried until the ShMS acknowledges them. This ensures that no hot-swap event is lost. The IPMC's each have a 50 message buffer, and the ShMC has room for an additional 10 messages. The chance of overrunning the message buffers is low, but it is possible if every blade in the chassis simultaneously generates events from multiple sensors.

Message corruption can happen on the IPMB under limited circumstances, such as when the IPMC is receiving and formatting the message due to a local IPMC error. If the message is corrupted, the IPMC checks both the header and the message checksum. If either the header or checksum is incorrect, the message is dropped. The sending IPMC then retries the message. There is little chance of the same message getting corrupted on the bus three times in succession.

If IPMB-0 Isolation events are seen frequently in Event Log

The following events might have a similar cause:

- A FRU sends frequent IPMB-0 Isolation events
- The active SCM reports frequent IPMB-0 Isolation events
- The ShMS is unable to communicate with shelf IPMCs because of frequent IPMB-0 Isolation issues in the shelf

The user can do several things respond to a persistent IPMB isolation issue when there is a shelf wide IPMB isolation alarm that is always reported by the active SCM. Perform the following steps:

1. Read the IPMB sensor (sensor 0x1100) on the active SCM. If the reading is 0xF8 or 0x8F then it indicates that the SCM is detecting an external fault on the bus. This means that some other IPMC on the bus is causing the IPMB issue. An 0xF8 or 0x8F reading also indicates that only one IPMB is down. 0xFF would indicate that both IPMBs are down.
2. Read the IPMB sensor on all the boards as well. The reading will provide data on what is causing the IPMB isolation. Refer to the PICMG specification for information on what the IPMB sensor reading means. Again, reading a '8' in only one of the nibbles in the IPMB-0 sensor reading indicates that the FRU is communicating on just one bus while it is isolated from the other. A reading of 0x88 indicates that the FRU is communicating on both IPMBs and there are no issues.
3. If only one IPMB is down (as reported from IPMB-0 sensor readings in steps 1 and 2) try isolating boards in the chassis, one by one, from that IPMB, as follows:
 - a. This can be done via HPI using the IPMB controls on each board's Resource. These controls are numbered 0x1101 for IPMB-A and 0x1102 for IPMB-B. If the IPMB-0 sensor on the active SCM reads 0x8F then IPMB-A is down and if it reads 0xF8 then IPMB-B is down. To force-isolate a board off IPMB-A you need to use control 0x1101 on the Resource for that board (e.g. Resource 0xA008C for a board in slot 10). For IPMB-B, use control 0x1102.
 - b. To isolate a board off an IPMB you can alternatively use the PlatformMgmt CLI as follows (your input is shown in bold):


```
platform-mgmt# front <N - slot number of board>
platform-mgmt(front N)# # configure control <0x1101/0x1102>
Current Control Details :
Control Type : ANALOG
Analog : 0
Enter new value for the control :
Value [0]: 0
Successfully set control.
Newly set Control Details :
Control Type : ANALOG
Analog : 0
platform-mgmt(front N)#
```

At this point the IPMC on the board should have isolated itself off the intended IPMB.
 - c. After doing this for each board, wait for 60 seconds and check to see if the IPMB isolation alarm is gone from the active SCM before moving on to the next board. If an IPMC is connected to the IPMB even when it is in an error state then force isolate it from that IPMB to resolve the issue. This should also help identify the board that is responsible for this issue. Then run some dedicated IPMB stability tests with just that board and the SCM-s in the chassis.

- d. To return the board to its default state where it is joined on that IPMB you can again use the CLI as follows:

```
platform-mgmt(front 1)# configure control 0x1101 auto
Default Control Details :
Control Type : ANALOG
Analog : 0
platform-mgmt(front 1)#
```

4. If the above procedure does not reveal the board that is causing the IPMB issues, next try resetting the IPMC of each of the boards in the shelf, one by one, and check after each board reset to see the IPMB isolation alarm is gone from the active SCM. Resetting the IPMC should reset all I²C interfaces on that IPMC and cause it to recover from any internal software errors. You can reset an IPMC via the rsys-ipmitool on the active SCM as follows:

```
rsys-ipmitool -I lan -H 127.0.0.1 -A NONE -t <IPMB Addr of board> raw 6 2
```

The IPMB addr of each slot is listed in the Shelf Manager reference manual. It is also the least significant byte of the ResourceId for that board (e.g. 0x1009A for board in slot 1)

5. If both of the above procedures do not work then most likely there is a hardware issue with one of the boards in the shelf. At this point physically extracting the boards from the chassis one by one would be the best way to go. Physically remove one blade at a time from the shelf and wait for at least 120 seconds to check if the IPMB Isolation alarm disappears. If not, then move to the next FRU.

Note: Only the front blades are mentioned above, but if the front blades have all been worked on and that still does not resolve the issue, the user should then try isolation on the PEMs and fans in that order.

Troubleshooting FRU and shelf operation issues

This section explains how to respond to FRU and shelf operation alarms and other issues. The Shelf Manager may be able to resolve some of these issues automatically, but if a problem continues, you may need to intervene.

FRU information layout

The FRU information storage locations use a common layout to support platform management tasks. The following general layout has mandatory and optional content requirements:

Information Area		Comments
Common Header		Mandatory
Internal Use Area		Optional
Chassis Information Area		Optional
Board Information Area		Optional
Product Information Area		Optional
MultiRecord Area:	Record 1	Optional
	Record 2	Optional
	...	Optional
	Record <i>n</i>	Optional

The common header is present on every board's FRU data record as well as on the shelf FRU data record. The board and shelf data records both support the Board, the Product Info, and the MultiRecord areas. The shelf FRU data additionally has a Chassis Information Area that contains the shelf serial number and manufacturer information.

Responding to specific alarms

Responding to a communication lost alarm

The Shelf Manager is not getting keep-alive responses from the target FRU, usually because the FRU was extracted without going through the normal hot-swap extraction procedure. Try these steps:

- If the FRU has been removed, the FRU can be removed from the RPT using HPI control 0x101e on the Shelf Manager resource, the Failed Resource Extract control. This also removes the alarm. See the *SAF Mapping Specification* for details.
- Initiate a discovery operation from the platform-management CLI using the `rediscoverShelf` command. Wait 60 seconds after the discovery completes to see if the alarm is removed.
- If the FRU is still in the shelf, physically extract and re-insert the FRU according to the procedure for removing and replacing a module in the *Platform Reference* manual.
- If the condition still persists, a firmware lockup or other hardware problem may have occurred in the IPMC or MMC. However, the full set of FRUs to which communication has been lost should be analyzed for patterns. The analysis may indicate a problem affecting multiple FRUs or affecting a hub module's ability to communicate with all other FRUs.

If the condition persists, contact Radisys Technical Support for assistance.

Responding to an IPMB disabled alarm

The interface between the IPMC and the external bus has been disabled by the IPMC.

1. If a FRU loses contact with the IPMB, query the FRU's IPMB sensor to determine the reason.
2. If the entire IPMB is down, the Shelf Manager module generates an alarm. Query the module's sensor to determine the reason.

To interpret the values of the physical IPMB-0 sensors, see the *AdvancedTCA Base Specification*. If the condition persists, contact Radisys Technical Support for assistance.

Responding to a Resource Failed alarm

The following symptoms apply to this alarm:

- A FRU enters the Resource Failed state and a ResourceFailed alarm is added to the Domain Alarm Table.
- The FRU may be periodically entering and leaving the Resource Failed state as well.

There are no hardware presence detect lines in ATCA backplanes. The ShMS thus relies on message based pings to periodically query each intelligent FRU whose presence is known to the ShMS, as follows:

1. To the IPMC of every intelligent FRU in the shelf the ShMS sends a `GetDeviceId` ping every 3 seconds by default. If a ping times out with a 0xC3 IPMI response code the ShMS schedules a second ping for that IPMC after 2 seconds.
2. If the second `GetDeviceId` ping also fails then the ShMS generates the `Mx->M7` event for the FRU where `x` is not 1, but it is the last known good hot-swap state of the FRU. This is the communication lost event which is mapped to a `ResourceFailed` condition in HPI. You will see the `ResourceFailed` alarm for that FRU's resource when this happens.
3. If `x` (last known good hot-swap state) is `M1` then a `M7` event is not generated. In this case the ShMS will deem that the FRU has been extracted from the slot and will generate an `M1->M0` event. This maps to a `INACTIVE->NOT_PRESENT` hot-swap event in HPI and the resource for that FRU will be removed from the HPI RPT after this event.
4. Following the `M7` event, the ShMS keeps trying to ping the IPMC that has lost communication every 3 seconds anyway to check if communication is regained. If the ping succeeds, the ShMS generates an `M7->Mx` event where `x` was the last known good hot-swap state. It then launches a rediscovery of that FRU to learn the current state of that FRU because it may have undergone changes while communication with the ShMS was lost.

There is a configuration parameter available for the Shelf Manager that can change the ShMS's presence ping intervals from the default of 3 seconds. This parameter is called `SHMS_FRU_HEARTBEAT_FREQUENCY` and can be set in the `/etc/shmgr.conf` file. You will need to reboot the SCM or restart the Shelf Manager after making this update. By default this parameter is set to 3 seconds in the read-only `/usr/share/shmgr/shmgr.defs` file which houses all the Shelf Manager parameters with their defaults.

A brief description of the FRU heartbeat parameter from the `/usr/share/shmgr/shmgr.defs` file:

```
# Set the default FRU presence ping frequency in seconds
# This is the frequency in which the shelf manager will ping
# intelligent FRU-s for their presence.
# Minimum is 2 secs, Maximum is 60 seconds, Default is 3 secs.
```

In order to change the FRU heartbeat parameter in the `shmgr.conf` file, insert the line:

```
SHMS_FRU_HEARTBEAT_FREQUENCY=<ping interval in secs>
```

Note: This is not a FRU-specific parameter. The new ping interval is global and applied to all intelligent FRUs in the shelf. A side effect of increasing the ping interval would be that real `M7` conditions for FRUs would get detected later than they are now (after about 5 seconds). Also once a FRU is physically extracted from the shelf the Shelf Manager would take longer to generate the `INACTIVE->NOT_PRESENT` event.

After making any shmgr.conf file update, remember to synchronize it to the standby SCM from the platform-management CLI before rebooting the SCMs. The procedure from the active SCM is:

```
mcli
platform-mgmt
shelf-mgmt
configure sync_config
```

Responding to a ResourceFailed alarm

The event SAHPI_RESE_RESOURCE_FAILURE is received both when a FRU is extracted too quickly (that is without the hot-swap LED powered on) and when there is an IPMC failure.

Because there is no “hardware presence detect” method for boards in ATCA shelves it is difficult for the Shelf Manager to distinguish between a surprise extraction and a real IPMC failure. One difference between a surprise extraction and an IPMC failure case is in how the 'recovery' event is received for the two cases, as follows:

1. For a surprise extraction-induced RESOURCE_FAILED state there is no RESOURCE_RESTORED event. When a new FRU is inserted in the same slot, you directly receive a last known good hot-swap state of “NOT_PRESENT” (e.g. ACTIVE->NOT_PRESENT) event for the failed resource with the event's severity set to the resource's severity (SAHPI_MAJOR for Front Boards). This is followed with the NOT_PRESENT->INACTIVE event for the resource for the newly inserted FRU in that slot.
2. For an IPMC failure-induced RESOURCE_FAILED state you receive a RESOURCE_RESTORED event when the IPMC recovers from the failure. There are no hot-swap events received here. IPMC failure induced resource failures are highly unlikely. Radisys IPMCs use a Watchdog run by the IPMC FPGA. The IPMC strobes the FPGA-maintained Watchdog periodically. If the IPMC fails, that watchdog expires and the FPGA resets the IPMC to recover it from its failure. IPMCs from most vendors employ similar logic to protect against unrecoverable IPMC failures. Chances of the IPMC failing continuously for more than 5 seconds (the time it takes for the ShMS to ping the IPMC twice for communication test) is very low. Of course an IPMB failure can cause a RESOURCE_FAILED condition that can recover later if the IPMC can recover itself from the IPMB error.

The next event from a ResourceFailed event can provide some information on whether the FRU was surprise extracted or simply lost communication for a time while still being present in that slot.

You can choose to simply mark a FRU as failed when you get the ResourceFailed event and failover any critical service running on that FRU to another FRU. Marking it as extracted might provide incorrect information to an operator in case the FRU is not actually extracted from the shelf. From a management perspective the FRU can be treated like it is INACTIVE so no services are assigned to it. Then when the next event comes (either a NOT_PRESENT hot-swap event or a RESOURCE_RESTORED event) you can take appropriate action by marking the FRU as either extracted (for NOT_PRESENT event) or reactivated (RESOURCE_RESTORED) event.

Responding to an IPMB isolated alarm

The module's IPMC has detected a problem on one or both of its IPMB interfaces, and has isolated the IPMC from the external bus.

1. Verify the module is seated properly.
2. Use the hot-swap procedure to extract and re-insert the module.

If the condition persists, contact Radisys Technical Support for assistance.

Responding to a fruinfo fail alarm

The *fruinfo* startup script failed to retrieve the information it needed.

1. Check for the Shelf FRU Invalid alarm, which indicates that the Shelf Manager cannot read the Shelf FRU data, or that the data is not valid. If the alarm is present, see [Shelf Manager module boots up in debug mode on page 126](#).
2. Verify the shelf address is in the Radisys format, as specified in [Radisys shelf address format on page 65](#).
3. Verify the shelf IP address is a valid address.

If the condition persists, contact Radisys Technical Support for assistance.

Responding to a PEM absent alarm

The Radisys shelves are designed to operate with two redundant power entry modules (PEMs), but can operate with only one PEM installed. However, because some FRUs specifically monitor voltages from each PEM individually, those FRUs generate alarms without both PEMs installed. Even if these alarms are acknowledged to silence the audible alarm, they may be reasserted at any time because the power line is floating.

If the shelf is to be run with only one PEM for any length of time, it may be desirable to disable event generation from the voltage sensors on those FRUs for the PEM that is removed. This may be done using the HPI function *saHpiEventEnableSet()*.

To disable event generation using *hpiapp*:

1. Access *hpiapp* from the Linux shell and list all alarms:

```
hpiapp
d
0
```

2. Note the resource ID of the FRU and the sensor ID listed in the alarm description.
3. Display the sensor information:

```
q
s
b
```

4. Verify that the sensor is related to the presence of a PEM.

5. Disable event generation on that sensor with these commands:

```
8
```

```
<resourceId>
```

```
<sensorId>
```

The screen displays the final message:

```
SensorEventsEnabled : 0x0
```

6. Exit *hpiapp*:

```
q
```

```
q
```

Responding to a redundancy lost alarm

Note: This section applies only to shelves where an SCM is acting as the Shelf Manager. It does not apply to ATCA-6002 shelves.

An alarm indicates that the Shelf Manager has not detected a second, redundant Shelf Manager. The alarm is generated by the Shelf Manager redundancy status sensor on the Shelf Manager resource. Try these steps to resolve the problem:

1. Log into each SCM and verify that the banner does not indicate that the SCM is in debug mode. If you do see a debug mode message, see [Shelf Manager module boots up in debug mode on page 126](#).
2. Verify that both SCMs are installed in the shelf's hub slots and that both have the hot-swap state of ACTIVE. Use the platform-management CLI command `show chassis` with the `information` option.

Syntax: `show chassis { status | details }`

3. Verify the Shelf Manager status and redundancy status using the shelf management mode command `show all`. Perform this command on both SCMs. If the results conflict, the results from the active SCM should be trusted.

Syntax: `show all`

4. Verify that the SCMs have correct IP addresses assigned to the dedicated link between the Shelf Managers. Log into each SCM, and execute the command:

```
ifconfig eth1
```

5. Look for the “inet addr” value. The SCM in the lower-numbered slot in the chassis should have an eth1 IP address of 10.0.1.1. The SCM in the higher-numbered slot should have the address 10.0.1.2. If the addresses were changed from these values—even in the template files that set the addresses upon SCM bootup—restore them to the original settings persistently. See the *Software Guide* for more information on the IP address assignment and template files.

If you are operating with only one SCM installed for a length of time, you may want to disable the redundancy lost alarm. To disable the alarm, use the same procedure described in [Responding to a PEM absent alarm on page 114](#).

Responding to a redundancy degraded event

Note: This section applies only to shelves where an SCM is acting as the Shelf Manager. It does not apply to ATCA-6002 shelves.

A SAHPI_ES_REDUNDANCY_DEGRADED event indicates that the Shelf Manager has detected a redundant Shelf Manager, but it is running a different software version. The event is generated by the Shelf Manager redundancy status sensor on the Shelf Manager resource. The new Shelf Manager features that are not shared by both Shelf Managers are not available. To resolve the problem, upgrade the LMP software to make the Shelf Manager versions match.

Responding to a shelf FRU invalid alarm

The shelf itself has a sensor to indicate that the Shelf Manager cannot read the Shelf FRU data, or that the data is not valid. If the sensor generates an alarm, see [Shelf Manager module boots up in debug mode on page 126](#).

Responding to a FRU stuck in an INSERTION_PENDING hot-swap state

A FRU may become stuck in the INSERTION_PENDING hot-swap state after one of the following events:

- Shelf PowerUp
- After inserting it in the shelf
- After power cycling the FRU programmatically

Before addressing the possible cause and trying to correct it, some of the information from the INSERTION_PENDING state may be useful. The HPI INSERTION_PENDING state internally has the following two stages:

1. The first stage is where the FRU is functionally INACTIVE. This is the ATCA M2 state (PICMG 3.0 ATCA specification). This is the stage where the FRU is waiting for the Shelf Manager or system manager to verify its presence in the shelf and 'push' it to the second stage. This is where the FRU automatically transitions to after chassis power-up if its latch is closed. Once ShMS gets the notification from the FRU's IPMC for transition to M2 state, the default action is to transition the FRU automatically to the next ATCA hot-swap state.
2. The second stage of INSERTION_PENDING is the ATCA M3 state where the Shelf Manager takes over and starts the power budgeting and E-Keying process for the FRU. The Shelf Manager grants power and applicable E-Keys to the FRU in this stage. At the conclusion of this stage, the FRU should transition to the ACTIVE state (ATCA M4 state).

Details on the ATCA hot-swap process are in the PICMG 3.0 R3.0 specification. Details on how HPI maps the ATCA hot-swap state machine to HPI hot-swap states are in the *SAF HPI-to-ATCA Mapping Specification*.

If the “FRU stuck in an INSERTION_PENDING hot-swap state” issue is seen, perform the following steps:

1. First determine the ATCA M state the FRU is in as listed for the hot-swap sensor of the FRU. This sensor is accessible in the FRU's sensor list (on the HPI Resource record for the FRU). For the hot-swap sensor number for a FRU, refer to the FRU's reference manual. The event state of the hotswap sensor reports the current ATCA M state of the FRU. For example, 0x04 indicates M2 and 0x08 indicates M3.
2. If the FRU is in M2, check the HPI Domain AutoInsertion timer to determine if it is set to blocking (-1) or to a large time value. Use hpiapp on the active SCM as follows:

```
hpiapp -> h -> 4
```

If the AutoInsertion timer (which is domain specific) is set to -1 then all FRUs will stay in ATCA M2 (HPI INSERTION_PENDING state) until the system manager programmatically activates the FRU via the HPI hotswap API `saHpiResourceActiveSet()`. If the AutoInsertion timer is set to a large time value (e.g. 60 secs), then the ShMS will keep the FRU in the INSERTION_PENDING state for 60 seconds before activating it. The user has the option of using the HPI hotswap API to activate the board manually before that timer expires or even to deactivate it.

3. If the FRU is found to be in M2 and AutoInsertion timer is not -1 or a large value, check the state of the 'FRU Activation Control' 0x1020 on the Slot Resource associated with the FRU. For example if the FRU is in slot 2 in a 14 slot shelf with ResourceId 0x20096 then its slot Resource will be 0x2FF96. You can determine the slot resource from a Resource's entity path. The FRU Resource's Entity Path will be a superset of its associated Slot Resource's entity path. For example, for the slot 2 board example above:

```
Slot Resource 0x2ff96 entity path -
/ADVANCEDTCA_CHASSIS_00/PHYSICAL_SLOT_02
FRU Resource 0x20096 entity path -
/ADVANCEDTCA_CHASSIS_00/PHYSICAL_SLOT_02/PICMG_FRONT_BLADE_02
```

If the 'FRU Activation Control' reports a MANUAL mode, it means that slot is configured to always be manually activated. As a consequence, the Shelf Manager will not activate the FRU, but instead leave it in an INSERTION_PENDING state. The system manager needs to manually activate the FRU in this case using the HPI hotswap API.

To restore automatic ShMS-controlled activation set the mode of the 'FRU Activation Control' to AUTO mode.

4. If the FRU is in M3, check to see if the Shelf Manager has granted it power. This can be checked via the 'Assigned Power Sensor', Sensor 0x1011, on the Slot Resource associated with the FRU. The mechanism for identifying the Slot Resource was explained under [Step 3](#) above.

A non-zero reading reported by the 'Assigned Power Sensor' indicates that the ShMS has granted power successfully to the FRU. Now it is the IPMC's responsibility to transition the FRU to M4. If the FRU is a Radisys board, contact Radisys Technical Support for further investigation.

5. If the FRU is in M3 and does not have payload power granted to it, the FRU might be asking for more power than the ShMS can grant to that slot. Perform the following steps:
 - a. Check the maximum power allowable to that FRU's slot by reading the 'Max Power Capability Sensor', sensor 0x1012, on the FRU's Slot Resource. Check how much power the FRU is requesting by reading the 'Desired Power - in tenths of Watts' control, 0x1030, on the FRU's Resource. If the FRU's requested power is more than the maximum power the ShMS can allow to that slot, that FRU cannot be activated in that slot.
 - b. If this FRU is a managed FRU (e.g. AMC), its requested power, the parent FRU's power, and any other managed FRU already activated on that parent FRU is possibly exceeding the maximum power capability of the parent FRU's slot.

For example, a slot might allow a Maximum of 200W. It already has an AMC Carrier and an RTM with power of 120W (including management power) and 30W respectively. In this case 150W is already granted from that slot's maximum budget. Now if an AMC is installed on the AMC carrier with its own requirement of 60W then the total power budget of that slot is exceeded and the ShMS would not be able to grant power to that AMC.

- c. Check the 'Desired Power' controls on all FRUs installed on that slot (parent and managed FRUs) and check if the total exceeds the maximum power capability of the slot.
6. If the FRU is in M3, the checks suggested in [Step 5](#) above all pass, and the FRU along with its managed FRUs is below the slot's maximum power capability, it is possible that the Shelf's total power capability is exceeded. In this case, collect all Shelf Manager logs from the /tmp/shmgr/ folder and forward them to Radisys Technical Support for further investigation.

Responding to a FRU not powered properly

Perform the following steps to confirm/correct this indication:

1. Check that the FRU is inserted correctly and that the hot-swap latch is in the closed position.
2. Check the blue hot-swap (H/S) LED on the FRU that is not powered correctly. The LED should be blinking with short blinks if it is waiting to be activated by the Shelf Manager. The LED should be off if the Shelf Manager is activating or has already activated the FRU.
 - a. If the hot-swap LED remains blinking for a long time, check the hot-swap state of the FRU. For details, see [Overriding the default ShMS hot-swap mechanism on page 72](#).
 - b. If the hot-swap LED is off, verify the FRU's voltage sensor readings are within specifications. If they are outside the specifications, remove and re-insert the FRU.
3. Check that IPMI communications to the FRU are working properly.
4. Use the hotswap CLI command to virtually extract and re-insert the FRU.

Responding to power LED off in SDP

Indications are that the I/O expander on the SDP that controls the GPOs for the alarm LEDs is getting reset. Other than the Power LED, all other alarm LEDs (including the audible alarms) are periodically updated by the Shelf Manager, so they will all recover to their correct states following that occurrence. However, the Power LED is never touched after ATCA-5014 IPMC initialization so if it turns off in runtime then it will never get updated again. That is why it remains off.

In newer ATCA-5014 firmware, an internal state machine constantly monitors the GPOs on the I/O Expander on the SDP and if any are found in an unexpected state such as Power LED off, it is recovered immediately. This kind of local recovery mechanism on the IPMC itself guarantees that the LEDs on the SDP synchronizes with the ATCA-5014.

Responding to a fan not running

1. Check that the Shelf Manager can communicate with the fan and that it is in the active state. Use the CLI command `show chassis status`.
2. If the fan is not in the active state, check that the hot-swap switch is closed.
 - a. On the ATCA-6000 shelf, the fan should have its hot-swap switch in the leftmost position.
 - b. On the ATCA-6006 shelf, the hot-swap latch thumbscrew must be secured.
 - c. On the ATCA-6014 and ATCA-6016 shelves, the H/S button must be toggled to the “on” state.
3. If the fan is still inactive, follow the instructions given in [Responding to a FRU not powered properly on page 119](#).

Responding to fan speeds higher than normal

1. Check that no temperature alarms are currently present. See [Responding to alarms on page 100](#).
2. If temperature alarms are present, check the actual value of the sensor that is in alarm. See [Investigating sensor values on page 102](#).
3. Check that the Shelf Manager can communicate with the fan and that it is in the active state. Use the CLI to access the fan slot and run the command `show` with the *details* option.

Syntax: `show { information | details | events | sensors { all | sensorId <id> } }`

If the Shelf Manager is communicating with the fans, it should reduce the fan speed when temperatures return to normal.

4. Check the air temperatures in the shelf.
5. Verify that the air filters have been cleaned according to the maintenance schedule.

If a FRU is generating a temperature alarm that doesn’t clear itself over time, this may indicate a hardware problem.

Troubleshooting Shelf Manager issues

General issues

When you encounter problems related to shelf management, check the following items first:

Make sure the Shelf Manager module is installed properly

Make sure the module running the Shelf Manager software is in the Radisys shelf and seated properly.

Make sure a proxy FRU is installed properly

Make sure at least one proxy FRU is in the Radisys shelf and seated properly. The proxy FRUs for each shelf are:

ATCA-6000: The SPMs located in the rear slots behind the SCMs

ATCA-6006: The PEMs located at the rear of the shelf

ATCA-6014 and ATCA-6016: The RCMs located in the front horizontal slots at the bottom.

Important: Do not remove both proxy FRUs at the same time.

Make sure the Shelf Manager module's ejector latches are fully closed

Make sure the Shelf Manager module's ejector latches are closed fully and that the blue hot-swap LED is off.

- If the latches are closed but the blue hot-swap LED is flashing, shelf management may have been disabled at the factory. Check with your Radisys representative if you suspect this.
- If the blue hot-swap LED is on, the latches may not be seated properly. Try opening the latches and then closing them again. The blue hot-swap LED should flash for a few seconds after the latches are closed before turning off.

Make sure the Shelf Manager software is running

Make sure the Shelf Manager software is running. The Active LED of the SCM (or one SCM of a redundant pair) should be amber and the OOS (Out of Service) LED should be off. For a module running the Shelf Manager in an ACTA-6002 shelf, the PWR LED should be green. In this shelf, the OOS LED is not updated by the Shelf Manager.

Additionally, the Linux command `shmgr status` should show a status of “started.” The `shmgr role` command should show “active” or “standby.”

- If the status is “stopped,” ensure that the SCM is in one of the two hub slots. The hub slots are 7 and 8 in the ATCA-6000 and ATCA-6014 shelves, 1 and 2 in the ATCA-6006 shelf, and 8 and 9 in the ATCA-6016 shelf.
- For the ATCA-6002 shelf, if the status is “stopped,” ensure that the module running the Shelf Manager is in node slot 1.
- If the status is “stopped,” shelf management functionality may have been disabled intentionally to let it operate in a non-Radisys shelf. If it was disabled intentionally, see the *Switch and Control Module Installation Guide* for instructions on re-enabling it. If it was not intentionally disabled, use the Linux command `shmgr start` to start it.
- If the role is “unknown,” Shelf Manager communication with the local IPMC is not working.
 - a. Restart the Shelf Manager and check the role of the Shelf Manager module with the `shmgr role` command again.
 - b. Reboot the Shelf Manager module and check the role again.

When the shelf has redundant SCMs, also make sure the Shelf Manager is running on the other SCM.

Check for IPMI serial device errors

Check for IPMI serial device driver errors as follows:

1. From the SCM’s Linux prompt, issue the command:


```
cat /proc/driver/ipmi
```
2. Verify that the “Rx” error counters at the end of the list are either zero or are not incrementing at regular intervals. Steadily increasing error counts here may indicate a hardware defect.

Blade and platform management issues

The topics in this section include ShMC-to-IPMC communications, SEL location, sensor event reporting, and IDR, HPI, and IPMI topics that address typical questions.

ShMC-to-IPMC communications

The ShMC and IPMC communicate with each other on the following occasions:

- Hot swap. At each occurrence according to the sequence of the events.
- Events and alarms. Immediately when the event is noted.
- Presence detection. Every three seconds.

In addition, the ShMC and IPMC will communicate whenever:

1. The Shelf Manager needs to retrieve data from the IPMC in order to activate it (FRU data, SDR-s, power requirements, E-Key data, and so on.).
2. The Shelf Manager needs to set data on the IPMC (e.g. update alarm indicators).
3. The user makes a request to retrieve data directly from the IPMC (for example, sensor readings).

All of these queries will go through the ShMC. If the Shelf Manager is not able to persistently communicate with an IPMC and all its queries fail with Timeout response codes (0xC3) then the IPMC will be set to the Communication Lost (M7) state. This will cause a HPI Resource Failed event to get generated for the IPMC's FRU and the user will get notified that communication with the IPMC is no longer available over the IPMB-0.

ShMC watchdog timer

The local management processor (LMP) sends a reset signal to the watchdog timer of the ShMC every 15 seconds. The ShMS probes this timer every two seconds.

Event log

Each IPMC maintains its System Event Log (SEL) in NVRAM that is local to the IPMC. The NVRAM is part of the "IPMC entity" circuitry that includes the IPMC FPGA as well. So the IPMC device itself, the IPMC FPGA, and NVRAM together constitute the IPM Controller (IPMC).

Because the SEL is kept in the NVRAM, it is persistent over board resets, power cycles, and IPMC resets as well as board extraction/reinsertion events.

The SEL can be read by standard IPMI commands defined in the IPMI specification. Tools such as `rsys-ipmitool` are available to read the SEL in human readable format. For example, to read the SEL of a board in slot 3 (IPMB addr 0x92) from the active SCM in slot 7, issue following command:

```
rsys-ipmitool -I lan -H 127.0.0.1 -A NONE -t 0x92 sel list
-- this prints the SEL in concise mode
rsys-ipmitool -I lan -H 127.0.0.1 -A NONE -t 0x92 -v sel list
-- this prints the SEL in verbose mode
```

Typically each IPMC has only its own events in its SEL. The ShMCs are exceptions to this rule because they log events from all IPMCs in the shelf in their SEL. As a consequence, the ShMC SEL is quite large and has nearly 19KB (about 1200 entries) of SEL space.

Shelf Manager tables

The Shelf Manager uses tables and files to store information and data related to events, alarms, sensors and resources, etc. The following user- or application-accessible files and tables exist in the Shelf Manager:

- `shmgr.log`
- Domain Event Log (DEL)
- System Event Log (SEL)
- DRT, RPT, RDR, DAT

The above list includes all the HPI accessible tables that users can deal with in their software. The Shelf Manager keeps many internal tables to manage the system, so tables of IPMCs, FRUs, SDRs, E-Keys, Fans, power supplies, etc. are maintained by the Shelf Manager. These are inaccessible to the user and are only for the shelf management software's internal usage. The tables are all maintained in RAM on the SCM LMP, and are only active while the Shelf Manager is running. Once the Shelf Manager is stopped or the SCM is reset, the inaccessible tables get cleaned up automatically.

Each table has a set size/event limit. For example the Shelf Manager can support up to 100 IPMCs with 32 FRUs on each IPMC and up to 1000 SDRs per IPMC. There are no limits on number of E-Keys per IPMC because that number is already limited by the size of board point-to-point records that the IPMC can contain.

The HPI server can support a maximum of 1024 events in the Domain Event Log. This is configurable and the default is 255. The HPI server can support a maximum of 1024 alarms in the DAT. Out of these 1024, a maximum of 512 can be user alarms. The number of user alarms is configurable with the default limit being 50.

The HPI DAT and DEL are maintained in RAM so they are not persistent over SCM resets. The DAT and DEL are, however, synchronized between peer Shelf Managers, so even if the active Shelf Manager resets, it is still accessible through the peer, newly active, Shelf Manager.

The RPT is capable of supporting the maximum FRU limit that can be supported by the Shelf Manager plus all slots associated with them. This means that RPT can support 3200 FRUs.

The default Shelf Manager log size is 8 MB. Every time the log reaches the 8MB limit it is copied to a zipped backup copy to save space. The Shelf Manager keeps the first two Shelf Manager log instances (those created right after Shelf Manager start) and the last ten backup copies. Under normal operating conditions there should be about 3 hours of logging in each `shmgr.log` copy. This means that about 30 hours of logs should be available at any given time.

The `shmgr.log` file size is configurable up to a maximum of 16 MB.

Inventory Data Record (IDR) topics

Radisys can provide a tool called `fruupdate` that can be used to update existing FRU inventory data on IPMCs. Currently however adding new records to IPMC FRU inventory data is not allowed via HPI.

The `fruupdate` tool accepts a new FRU data image and config file as input. The new FRU data image must be created by the user with their custom FRU data records. The config file specifies the identity of those records. The tool retrieves current data from the IPMC and then applies changes specified by the config file to the retrieved image so a new updated image is created. The tool then writes the updated FRU data image back to the IPMC. The tool comes with documentation that the user can follow to create the new FRU data image and set up the config file.

HPI and IPMI topics

The “Platform Adaptation Layer” (PAL) transmits each request message three times at 50 sec intervals following a timeout response. Requests are only retried for timeout and busy error codes.

If an HPI client uses RMCP, the destination of the message from the client is the destination IP address of the active Shelf Manager and the destination port is 623 (RMCP port). Usually the user is expected to use the active Shelf Manager IP address (bound to the `dtl0:1` interface on the active SCM) for this purpose.

For some requests the HPI server and Shelf Manager can directly respond to the request without needing to retrieve information from the IPMCs in the shelf (e.g. `GetRptEntry`, `GetDomainInfo`, `GetEventLog`, and so on).

For others, like `GetSensorReading`, IPMI requests are sent via ShMC through the IPMB-0 to the IPMC. The IPMC responds to the request that the Shelf Manager translates to HPI-specific data and responds to the user.

Specific issues

[Table 15](#) lists specific issues that can be encountered while using the shelf management and HPI services on the Shelf Manager module.

Table 15. Symptom/response for Shelf Manager problems

Symptom	Response
Shelf Manager module boots up in debug mode Error code: Shelf FRU Invalid	<ul style="list-style-type: none"> Check for the following text after the DEBUG mode login banner: <pre>*** Boot error logged: *** /sbin/fruinfo failed to retrieve valid information from shelf manager</pre> This indicates that the Shelf Manager could not locate valid shelf FRU information. Proceed with the steps below. If the above text does not appear, see the <i>Software Guide</i> for help with other conditions that can cause the module to be in debug mode. Verify that the Shelf Manager module is installed in a Radisys shelf. The module goes into debug mode if its Shelf Manager attempts to start up in a non-Radisys shelf. See Disabling the Shelf Manager on page 37 for instructions on disabling the Shelf Manager. When the shelf has redundant SCMs, check whether both SCMs are in debug mode by logging on to the other SCM. If both are in debug mode, there is a problem with the shelf FRU information (see the next item). If only one is in debug mode, see page 127. Check the Shelf Manager log file (<i>/tmp/shmgr/shmgr.log</i> or the first two backup files) to determine if: <ul style="list-style-type: none"> The shelf FRU information is corrupted. Search for a message that starts with, "(Error) SHMS<FRU Data>:". If found, see Restoring shelf FRU information on page 132 to resolve the issue. The shelf FRU device is not accessible. Search for one of these messages, "Could not determine FRU size returning 0" or "Did not find a valid address". If found, see the next item to resolve it. If the shelf FRU device is not accessible, the proxy FRU for accessing the shelf FRU device may not be present or properly installed. Use these steps: <ul style="list-style-type: none"> Verify that one or more proxy FRUs are present (not applicable for ATCA-6002). The proxy FRUs for each shelf are: ATCA-6000: The SPMs located in the rear slots behind the SCMs ATCA-6006: The PEMs located at the rear of the shelf ATCA-6014 and ATCA-6016: The RCMs located in the front horizontal slots at the bottom Remove and re-insert the proxy FRUs (one at a time) and ensure they are seated properly. WARNING! Use safety precautions when you hot swap the PEMs on the ATCA-6006 shelf. See the <i>SYS-6006 Platform Hardware Reference</i> for instructions. Reboot the chassis and verify that the problem still exists. Remove and re-insert the modules and ensure they are seated properly. <p><i>(continued on next page)</i></p>

Table 15. Symptom/response for Shelf Manager problems (continued)

Symptom	Response
	<p><i>(continued from previous page)</i></p> <p>If the modules are still in debug mode, use these steps:</p> <ul style="list-style-type: none"> From each Shelf Manager module, verify that the shelf FRU device is accessible using this Linux command: rsys-ipmitool -t <IPMB> fru print 1 <p><SCM_IP> is the module IP address (which is usually 127.0.0.1), and <IPMB> is the IPMB address of the proxy FRU (see IPMB addresses of slots and FRUs on page 134).</p> <p>If an error occurs, the shelf FRU device may need replacement. Contact Radisys Technical Support. If the device can be contacted and information is shown, go to the next step.</p> Delete the copy of shelf FRU information (<i>/var/lib/shmgr/ShelfFruCopy.bin</i>) from each Shelf Manager module to prevent conflicts with the information in the shelf FRU device. Remove and re-insert the proxy FRUs and SCMs again. Reboot the shelf twice. If the problem persists, contact Radisys Technical Support. If only the standby SCM is in debug mode, communication between the SCMs over the backplane LAN synchronization interface (eth1) may be down. Avoid rebooting the active SCM when it is working normally and the standby SCM is in debug mode. Try as many of these steps as necessary: <ul style="list-style-type: none"> Verify that the update channel E-Key is enabled by checking the E-Keying Link State sensor. Use the platform-management CLI from the active SCM, and view the sensor on the active SCM. If disabled, remove and re-insert the standby SCM and ensure that it is seated properly. Verify that the eth1 interface is enabled on both SCMs using this Linux command: ifconfig eth1 <p>Verify that the interface is up and a valid IP address (usually 10.0.1.<logical_slot>) is assigned.</p> If necessary, enable eth1 on both SCMs by entering: ifup eth1 Remove and re-insert the standby SCM and ensure that it is seated properly. Call Radisys Technical Support if the condition persists. For other reasons the SCM could be in debug mode, see the <i>Software Guide</i>. Call Radisys Technical Support if the condition persists.

Table 15. Symptom/response for Shelf Manager problems (continued)

Symptom	Response
SCM resets abruptly forcing the LMP to reboot. There are no associated hot-swap events.	<p>Possible causes and steps to take are as follows:</p> <ol style="list-style-type: none"> 1. The SCM LMP hung, most likely because a process running on the LMP got into an error state that consumed all CPU cycles. This prevents the Shelf Manager from getting enough CPU cycles to strobe the IPMC Watchdog. The watchdog timer expires and resets the LMP. In this case a WATCHDOG event from the reset SCM should be found in the Domain Event Log of the active SCM. <p>Because it is difficult to identify the process that causes the LMP reset, periodically monitor the CPU usage of the processes running on the LMP to study the CPU usage pattern on that SCM while it is running after the reset. The linux 'top' utility is one way to get this data. If there is a process with occasional CPU usage spikes, this might be chronic condition that keeps happening but usually does not cause harm. For some reason that process' CPU usage spiked more than usual when the issue occurred, and that caused the WDT to reset. By collecting data even after the SCM has reset, it is possible to determine if such a process does exist. If so, Radisys technical support can do a more focused investigation on that specific process.</p> <ol style="list-style-type: none"> 2. The Shelf Manager (ShMS) that strobes the IPMI watchdog crashed and could not restart within 60 seconds. <p>If the ShMS has a catastrophic fault and crashes, it is immediately restarted by a keepalive process. The keepalive process first resets the IPMI Watchdog to 60 seconds, then copies over all logs and process data in persistent storage at /var/lib/shmgr/diagnostics and then restarts the Shelf Manager. The newly started Shelf Manager then restarts the IPMI watchdog strobe process. If for some reason the ShMS is not restarted within 60 seconds of it crashing, the watchdog will expire and reboot the SCM. A user can check if a ShMS crash did occur by checking the contents of the /var/lib/shmgr/diagnostics folder after the SCM has come back up after the reset. If files in that folder have a timestamp matching with when the SCM did reset, then a ShMS crash is confirmed. In that case, tar up the /var/lib/shmgr/diagnostics folder and send it to Radisys Technical Support for investigation.</p>
HPI Application cannot load HPI client library (HCL)	<p>Ensure the library was built using the <i>autobuild</i> package provided for the HCL in the host machine where the application is intended to be run. Also, install the HCL in the common library location of the host machine so the HPI application can find it during runtime.</p>
<p>Sessions cannot be opened on the HPI server.</p> <p>Error codes: SA_ERR_HPI_NO_RESPONSE or SA_ERR_HPI_INVALID_DOMAIN</p>	<ul style="list-style-type: none"> • Ensure the HPI Service is running on the SCM. Since the HPI server is integrated with the shelf management server, the same troubleshooting methods described in General issues on page 121 can be used here. • If the HPI application is being run from a remote location, ensure the HPI-specific environment variables are set up correctly and the correct DomainId is being used. <p>For opening sessions on the HPI service running on the active Shelf Manager, set the environment variable SAHPI_UNSPECIFIED_DOMAIN_ID to the common Shelf Manager IP address of the shelf where HPI management access is desired. The domain identifier SAHPI_UNSPECIFIED_DOMAIN_ID can then be used to open sessions.</p> <p>See Changing Shelf Settings Using HPI on page 53 for details.</p> <ul style="list-style-type: none"> • Ensure the network interfaces and routing tables of the host machine running the HPI application are configured correctly to communicate with the Shelf Manager over the network.
<p>Sessions cannot be opened on the HPI server.</p> <p>Error code: SA_ERR_HPI_BUSY</p>	<p>If this error is received following the shelf power-on, the HPI server is currently initializing and still going through the discovery process.</p> <p>During the shelf power-on reset, it might take some time (2–3 minutes) to discover all the entities present in the chassis, especially if the chassis is fully loaded. Wait for one minute and retry the session open request.</p>

Table 15. Symptom/response for Shelf Manager problems (continued)

Symptom	Response
Sessions cannot be opened on the HPI server. Error code: SA_ERR_HPI_OUT_OF_SPACE	The HPI server has reached the limit (32) for the number of sessions that can be opened simultaneously. Close a few sessions and try again, or use any of the already opened sessions for the necessary operations. Previous sessions may have been closed without issuing the <i>saHpiSessionClose()</i> function, which leaves the sessions in an orphaned state. Orphaned sessions are closed after 60 seconds, so wait 60 seconds and try again.
Sessions cannot be opened on the HPI server due to a response timeout. Error code: SA_ERR_HPI_NO_RESPONSE	An SA_ERR_HPI_NO_RESPONSE during a routine HPI operation usually maps to the IPMI 0xC3 (request timed out) response code. This is usually not expected but sometimes if the IPMC is busy responding to other requests it is possible that the response to a request can get delayed beyond the 250msec IPMI request timeout period which would result in the request timing out on the Shelf Manager side and a SA_ERR_HPI_NO_RESPONSE return code would be sent for the corresponding HPI request. The HSD retries each IPMI command 3 times so all 3 tries have to time out before the SA_ERR_HPI_NO_RESPONSE return code is sent back. In live operation this is very unlikely but can happen in specific situations. Following a Power-on operation initiation it is likely that the IPMC can get very busy because the ShMS will be reading SDRs and FRU data from it. Additionally Power-Budgeting and E-Keying can also occur while SDRs are being read. If IPMI requests to retrieve power-state (done by reading HS sensor) are sent during this time it is possible that the response can get delayed a bit because the IPMC is simultaneously handling a lot of other requests. Now one of the 3 tries would more often than not succeed but perhaps once in 1000 tries there can be a situation where all 3 retries are delayed and This results in an SA_ERR_HPI_NO_RESPONSE return code. Reissuing that HPI request should succeed.
A hot-swap cannot be completed because of a communications timeout. Error indication: The hot-swap LED does not start or complete its deactivation or activation sequence.	The M state of the ShMC/IPMC has no bearing on message timeout and retrial. The ShMC and other IPMCs all share the same messaging infrastructure. All requests originating from the ShMC/IPMC are retried three times at 250 msec interval each. If all three retries get no response then a timeout response code (0xC3) is sent back to the requester. The requester might just be an internal module of the IPMC (e.g. hot-swap state machine that generates hot-swap events), or it might be an application running on the LMP that sent the request through the ShMC (e.g. Shelf Manager). The requester might decide to retry the request again in which case the ShMC will do the same three retrials for the next request. In the case of hot-swap events, the IPMC generating the event will automatically keep retrying the event until it gets a success return code from the Shelf Manager. This guarantees that the Shelf Manager always receives hot-swap events from all IPMCs in the shelf, but it might result in a hot-swap deactivation/activation process hang due to no success return code.
RMCP sessions cannot be opened on the Shelf Manager.	<ul style="list-style-type: none"> Verify connectivity to the IP address by connecting with Telnet. RMCP sessions can only be opened on the active Shelf Manager. If using the IP address of an SCM, verify that it is hosting the active Shelf Manager. You can use the Shelf Manager IP address instead to ensure you reach the active Shelf Manager. The RMCP server may have reached the limit (32) for the number of sessions that can be opened simultaneously. Close a few sessions and try again. Or if possible, use any of the already opened sessions for the necessary operations.
Resource presence table (RPT) has only one resource entry present — ResourceId 0x1	This can happen if the Shelf Manager was not able to find a valid shelf FRU device and was thus not able to initialize the shelf. The RPT will then have only one resource associated with the shelf. Sensor 0x1000 (shelf FRU info valid sensor) in the ATCA shelf resource will generate an alarm to inform the user that the shelf FRU information is corrupt and needs to be reset. Following a reset of the shelf FRU information, power cycling the module hosting the active Shelf Manager should allow the Shelf Manager to fully initialize the shelf and thereby allow full resource discovery. If the problem is not resolved by a module reboot, power cycle the shelf.

Table 15. Symptom/response for Shelf Manager problems (continued)

Symptom	Response
RPT is incomplete; not all FRUs present in the shelf have associated Resources.	<ol style="list-style-type: none"> 1. Initiate a discovery operation using the CLI command rediscoverShelf, using the <i>saHpiDiscover()</i> function, or using the SNMP <i>saHpiDiscover.0</i> variable. This will initiate a fresh discovery of the shelf and create Resources for any FRU that was not already present in the RPT. 2. If the discovery fails to complete the RPT and Resources are missing for a FRU, try extracting that FRU and re-inserting it in the shelf. 3. The IPMI bus may be locked up or too busy. Try removing other modules from the shelf that may be causing IPMI bus errors. Then try running a discovery operation again. 4. Check if the FRU is in the Inactive hot-swap state (hot-swap LED is solid blue). The HPI server is not able to detect complete FRU information when in this state and thus cannot create an associated resource for it. Open and close the hot-swap ejector latch to activate full management access to the FRU, thus enabling the HPI server to create the resource.
Sensor list in the RDR of a FRU's resource is incomplete; not all expected sensors were found	Due to a communication error with the FRU during its activation phase, not all sensor data records (SDRs) could be loaded from it. Extract the FRU and re-insert it to freshly create the resource.
The LCD panel on the ATCA-6000 shelf's SDP is not updating, and there is no backlight.	<ul style="list-style-type: none"> • The HPI server might still be initializing and going through a discovery process. Open a session on the HPI server. If a SA_ERR_HPI_BUSY error code is returned, the server is still initializing and will start updating the LCD with system health information once initialization is done. • Check if there is a resource for the virtual SPM. The ResourceIds are listed in Table 13 on page 95. If the virtual SPM resource is not present, check for the individual SPM resources. If they do not exist, initiate a discovery operation using the CLI command rediscoverShelf, using the <i>saHpiDiscover()</i> function, or using the SNMP <i>saHpiDiscover.0</i> variable. If discovery fails to create Resources for the SPMs, extract and re-insert the SPMs. • If the SPM resource is present, communication between the SPM and SDP may have broken down. Send some custom text to the LCD using control 0x9 on the SPM resource and determine if the LCD gets updated with the text. If not, a shelf power cycle is required to re-enable the SDP LCD.
Cannot initiate hot-swap or power management commands	<p>Determine if the session from which the commands are being sent is opened on the SCM running the active Shelf Manager. Take the following steps to ascertain if the current HPI service is running on the active Shelf Manager:</p> <ol style="list-style-type: none"> 1. Use the <i>saHpiResourceIdGet()</i> function to find the resource upon which the current HPI service is running. 2. Then check the reading from sensor 0x1002 in the Shelf Manager resource (ResourceId 0x2) using <i>saHpiSensorReadingGet()</i> with SensorNum = 0x1002 and ResourceId = 0x2. <p>If the ResourceId from step1 does not match the sensor reading from step 2, then the current HPI service is not the active instance. Close the current session and open a new session on the active HPI server using the correct IP address of the active Shelf Manager as DomainID. See Chapter 4, Changing Shelf Settings Using HPI, on page 53 for details on opening sessions on the HPI server running with the active Shelf Manager.</p> <ul style="list-style-type: none"> • Determine if the resource is in the ResourceFailed state by checking the ResourceFailed flag in the RPT entry. Extracting and re-inserting the FRU in such a condition can help re-establish management access to it.
HPI commands on a particular resource are failing. Error code: SA_ERR_HPI_NO_RESPONSE	Determine if the resource is in the ResourceFailed state by checking the ResourceFailed flag in the RPT entry. If set to SAHPI_TRUE, the FRU associated with the resource is in a communication lost state with the Shelf Manager. Extracting and re-inserting the FRU in such a condition can help re-establish management access to it.

Table 15. Symptom/response for Shelf Manager problems (continued)

Symptom	Response
A user alarm cannot be created. Error code: SA_ERR_HPI_OUT_OF_SPACE	The DAT has reached the limit (50 by default) for the number of user alarms that can be active simultaneously. Consider these options: <ul style="list-style-type: none"> Delete the oldest user alarm from the DAT Delete the most minor user alarm from the DAT Choose not to add the new user alarm For information about configuring the number of user alarms, see Configuring HPI alarms on page 46 .
Data cannot be added to the MultiRecord area. Error code: SA_ERR_HPI_OUT_OF_SPACE	Neither new records nor new data for the existing PICMG MultiRecord area can be added directly to the FRU data inventory for a module or a shelf. The OEM MultiRecord area can be updated directly. Instead, use one of these two methods to update the MultiRecord area: <ul style="list-style-type: none"> Setting controls in the shelf resource Using a utility such as <i>rsys-ipmitool</i> to overwrite the existing image

Resolving shelf FRU information issues

This section provides information on resolving shelf FRU information problems, including how to save and restore shelf FRU information using the *rsys-ipmitool* utility, which is installed on the Shelf Manager module. The open source *ipmitool* utility can also be used. A list of IPMB addresses is included.

Actions to avoid

Shelf FRU information records can get corrupted when:

- Interruptions occur when writing to the shelf FRU device. Interruptions might occur if the Shelf Manager is stopped or the SCM power is removed while writing a record. They can also occur if the remote host from which you are updating the shelf FRU data using *ipmitool* is shut down or loses its connection to the SCM.
- An application writes directly to the shelf FRU device with invalid data.
- Both SCMs are swapped out at the same time in an ATCA-6006 shelf when the SCMs' persistent copy files are different and do not match the shelf FRU device. Swapping out a single SCM does not cause a problem.

Understanding what has occurred

If the active Shelf Manager cannot access the shelf FRU device, the Shelf Manager tries to verify its own information by getting information from standby Shelf Manager. The Shelf Manager will not start up with only its own copy of the shelf FRU information unless there is only one SCM in the shelf. Instead the SCM will go into debug mode.

Errors are written to the Shelf Manager logs when the data in the shelf FRU device cannot be read or overwritten.

Restoring shelf FRU information

The procedure below is useful when the shelf FRU information is bad and needs to be replaced in the shelf FRU device(s). This procedure applies to all Radisys shelves except for the ATCA-6002 shelf.

To write to the shelf FRU device:

1. Log in to any system with LAN connectivity to the SCM.
2. Locate a shelf FRU information image to copy to all needed locations and place it on the same system where you perform this procedure.

Possible sources of an image include:

- Cached copies that the active and standby SCMs keep in */var/lib/shmgr/ShelfFruCopy.bin*
- The file supplied in the Radisys software distribution (see the files for the shelves in the ATCA-6xxx directory)
- A file you have manually saved (see [Saving FRU information to a file on page 133](#))

3. If using the *ShelfFruCopy.bin* file on the SCM, rename or copy this file to another directory.
4. Delete the */var/lib/shmgr/ShelfFruCopy.bin* file from both the active and standby SCMs.
5. Write the image to the shelf FRU device through the proxy device using this command:

```
rsys-ipmitool -t <IPMB> fru write 1 <image>
```

<SCM_IP> is the IP address of the SCM (usually 127.0.0.1 when executed from the SCM itself).

<IPMB> is the IPMB address of a proxy FRU that gives access to the shelf FRU device. See [IPMB addresses of slots and FRUs on page 134](#) for addresses. Use the proxy FRU for your shelf model:

- For the ATCA-6000, use the IPMB address of the virtual SPM.
- For the ATCA-6006, use the address of a PEM. Each shelf FRU device must be updated separately (see [Step 6](#)).
- For the ATCA-6014 and ATCA-6016, use the virtual RCM address, which updates both shelf FRU devices.

<image> is the full path of the file (located on your current system) to be written to the shelf FRU device.

The shelf FRU device(s) and active and standby Shelf Managers are updated with the information.

6. On the ATCA-6006 shelf, perform [Step 5](#) again using the IPMB address of the other PEM.
7. Reboot both SCMs.

After the reboot, the active Shelf Manager retrieves the shelf FRU information from the shelf FRU device and writes new cached copies on both SCMs.

If this procedure does not resolve the problem, contact Radisys Technical Support.

Restoring shelf FRU information on ATCA-6002 shelves

The ATCA-6002 shelf uses a file to store shelf FRU information, because there is no physical shelf FRU device. This file is located here on the Shelf Manager module:

`/var/lib/shmgr/atca-6002.bin`

If the shelf FRU information gets corrupted, replace the *atca-6002.bin* file with the factory copy on the release CD.

Saving FRU information to a file

This procedure allows you to save either the shelf FRU information or information for an individual FRU to a file.

To save FRU information to a file:

1. Log in to any system with LAN connectivity to the FRU.

The file will be saved to this system.

2. Copy the FRU information to a file:

```
rsys-ipmitool -t <IPMB> fru read <FRU_ID> <target_file>
```

<IP> is the IP address of a Shelf Manager module (for example, this is usually 10.0.0.1 when executed from an SCM).

<IPMB> is the IPMB address of the FRU or of the proxy FRU that gives access to the shelf FRU device. See [IPMB addresses of slots and FRUs on page 134](#) for addresses. To save the shelf FRU information, use the proxy FRU for your shelf model:

- For the ATCA-6000, use the IPMB address of the virtual SPM.
- For the ATCA-6006, use the address of the PEM hosting the shelf FRU device.
- For the ATCA-6014 and ATCA-6016, use the virtual RCM address for the active shelf FRU device, or either RCM address for a particular device.

<FRU_ID> is:

- 0 for the information for a FRU and all devices for which it is proxy
- 1 for the shelf FRU device

<target_file> is the full path of the file created by copying from the shelf FRU device. The file is written to your current system and given the path and file name you specify (or put in your current working directory if none is specified).

If necessary, this saved image can later be restored as described in [Restoring shelf FRU information on page 132](#).

IPMB addresses of slots and FRUs

Table 16 shows the IPMB addresses of slots in the ATCA-6002 shelf. Table 17 shows the IPMB addresses of slots and FRUs in the remaining Radisys ATCA-6xxx shelves. The addresses are given in hexadecimal notation.

Table 16. Hex IPMB addresses of slots in the ATCA-6002 shelf

Slot	IPMB address
Active Shelf Manager	20
Front slot 1	82
Front slot 2	84

Table 17. Hex IPMB addresses of slots and modules in larger Radisys shelves

Slot or FRU	IPMB address / location			
	ATCA-6000 shelf	ATCA-6006 shelf	ATCA-6014 shelf	ATCA-6016 shelf
Active Shelf Manager	20	20	20	20
Front slot 1	9A	82	9A	9E
Front slot 2	96	84	96	9A
Front slot 3	92	86	92	96
Front slot 4	8E	88	8E	92
Front slot 5	8A	8A	8A	8E
Front slot 6	86	8C	86	8A
Front slot 7	82	n/a	82	86
Front slot 8	84	n/a	84	82
Front slot 9	88	n/a	88	84
Front slot 10	8C	n/a	8C	88
Front slot 11	90	n/a	90	8C
Front slot 12	94	n/a	94	90
Front slot 13	98	n/a	98	94
Front slot 14	9C	n/a	9C	98
Front Slot 15	n/a	n/a	n/a	9C
Front Slot 16	n/a	n/a	n/a	A0
PEM 1 (left PEM when viewed from rear)	C4	66	60, FRU ID 6	60, FRU ID 6
PEM 2 (right PEM when viewed from rear)	C6	68	60, FRU ID 7	60, FRU ID 7
Fan 1 (viewed from front)	C8 / Left top AMM	5A / Right fan tray	60, FRU ID 3 / Left fan tray	60, FRU ID 3 / Left fan tray
Fan 2 (viewed from front)	CA / Right top AMM	5C / Left fan tray	60, FRU ID 4 / Center fan tray	60, FRU ID 4 / Center fan tray
Fan 3 (viewed from front)	CC / Left recessed AMM	n/a	60, FRU ID 5 / Right fan tray	60, FRU ID 5 / Right fan tray
Fan 4 (viewed from front)	CE / Right recessed AMM	n/a	n/a	n/a
Virtual SPM (represents the active ATCA-5010 SPM)	50	n/a	n/a	n/a
SPM 1 ^a (rear slot 7)	E4	n/a	n/a	n/a
SPM 2 ^a (rear slot 8)	E6	n/a	n/a	n/a
Virtual RCM (represents the active RCM)	n/a	n/a	60	60
RCM 1 (left)	n/a	n/a	10	10
RCM 2 (right)	n/a	n/a	12	12

Table 17. Hex IPMB addresses of slots and modules in larger Radisys shelves (continued)

Slot or FRU	IPMB address / location			
	ATCA-6000 shelf	ATCA-6006 shelf	ATCA-6014 shelf	ATCA-6016 shelf
Shelf alarm panel	n/a	52	n/a	n/a

^a With the ATCA-5010 and ATCA-5014, use the virtual SPM address.

Resource types of slots and FRUs in the RPT

The resource presence table identifies the type of each resource in the 2nd byte of the ResourceId, as explained in [Radisys HPI implementation details on page 53](#).

The following are the Radisys-defined FRU resource types in hex (which correspond to the site types defined by PICMG):

- 00 All ATCA front modules
- 01 Power entry modules (PEMs)
- 02 Shelf FRU device
- 03 Dedicated ShMC
- 04 Fan tray
- 05 Fan filter tray
- 06 Alarm device (such as the shelf alarm panel on the ATCA-6006 shelf)
- 07 Advanced mezzanine card (AMC)
- 08 PCI mezzanine card (PMC)
- 09 Rear transition module (RTM)
- C0 Shelf peripheral module (SPM) or Radisys Chassis Manager (RCM)
- C2 Virtual SPM (represents the active SPM) or virtual RCM (represents the active RCM)
- C5 COM-E module

The following are the slot resource types in hex:

F0	Unknown slot
F1	SPM or RCM slot
F2	Shelf display panel slot
F3	Virtual SPM or virtual RCM slot
F6	RTM slot
F7	PMC slot
F8	AMC slot
F9	Alarm device slot
FA	Fan filter tray slot
FB	Fan tray slot
FC	Dedicated ShMC slot
FD	Shelf FRU slot
FE	PEM slot
FF	FRU slot
F5	COM-E module slot

Shelf Manager Initialization

Shelf Manager and HPI interface initialization

The shelf management server (ShMS) is installed as part of the local management processor (LMP) software image. The Shelf Manager Server daemon automatically starts as a background process at boot time using the Linux *init* script named *shmgr*. This is one of the first applications the kernel starts.

The following is the initialization sequence. This sequence applies to all Radisys shelves, including the ATCA-6002 when it is running in enhanced mode.

1. The *init* script loads default settings from */usr/share/shmgr/shmgr.defs*.
2. User configuration settings are loaded from */etc/shmgr.conf*. These override any settings in the defaults file.
3. The IPMI serial device driver, *ipmi_ser_mod.ko*, is loaded. This driver allows IPMI messages to be transferred between the LMP and its local Intelligent Platform Management Controller (IPMC). The IPMC implements a bridging function so that the host processor can also send IPMI messages to any other devices attached to the Intelligent Platform Management Bus (IPMB), such as node modules.
4. The Shelf Manager Server daemon, *shmgrd*, is started and initialized. If two SCMs are installed, the Shelf Manager Server also determines if it is functioning as the active or standby Shelf Manager.
5. The Shelf Manager Server loads and processes the contents of the shelf FRU device, which contains additional configuration data specific to the shelf in which the Shelf Manager is residing. Examples of shelf FRU configuration data include backplane channel connectivity, power sequencing information, and the number of front board slots.
Note: On the ATCA-6002 shelf, the shelf FRU data is stored in a file called *atca-6002.bin*.
6. For the active Shelf Manager only, the common Shelf Manager IP address is enabled, allowing management applications to connect to the Shelf Manager Server using Telnet and IPMI/LAN. The IP address information is also taken from the shelf FRU device.
7. The HPI server is started.
8. While the HPI server is initializing, the Shelf Manager then proceeds to apply power and grant E-Keys to all of the devices that have been detected in the shelf.
9. By gathering the information discovered by the Shelf Manager and also by querying intelligent managed devices directly as necessary, the HPI server does the following in the order listed:
 - a. Establishes a domain controller, an event management service and an alarm management service.
 - b. Initializes the domain event log (DEL) and the domain alarm table.
 - c. Initializes the resource presence table (RPT).

Shelf Manager Initialization

- d. Populates the RPT with resources and their associated resource data records (RDRs). Resources are created for each slot present in the chassis as well as all installed FRUs. RDR entries created include sensors (both physical and logical), controls, watchdogs, and inventory data records.
- e. Checks for the presence of previously saved HPI parameter control configuration files and loads them for each resource in the RPT. The HPI parameter control configuration files are archived in `hcf.tgz` in the `/var/lib/shmgr` directory. If no saved configuration files are present, the default configuration files are copied as the initial saved configuration.
- f. Starts the Event thread to dynamically listen for events from the Shelf Manager (only on the active HPI server) as well as from HPI applications.
- g. Starts the Session management service. At this point, the active HPI server is fully initialized and ready to accept Session open requests from external applications. However, the standby HPI server will still block session open requests until the completion of the next step.
- h. Starts the synchronization process on the standby HPI server to synchronize the active-standby HPI servers. Along with resource, sensor and control data, this process also synchronizes the HPI parameter control configuration files.
- i. This completes the initialization process for the standby HPI server, which can now accept session open requests from external HPI applications needing debug access to it.

Although the HPI server may take some time to become fully initialized when there are many FRUs in the shelf, the Shelf Manager becomes fully operational within 60 seconds of starting the shelf. The Shelf Manager runs in the background even as the HPI server is completing its initialization.

When shelf management services are not fully initialized on one of the SCMs, the red OOS (Out of Service) LED on that SCM is illuminated. As soon as the Shelf Manager is fully capable of managing the shelf, the LED turns off.

Shelf Manager Initialization

HPI subagent initialization

When the HPI subagent is started, initially you will see just the static scalar objects defined in the HPI-B0101-MIB module. One of these scalar objects is the saHpiDiscover.0 variable, which initially returns the value true(1).

Once the HPI server is running, the HPI subagent opens an HPI session to it and starts to dynamically populate all of the columns and rows in the HPI-B0101-MIB module tables. When this task is completed, the HPI subagent changes the value of the saHpiDiscover.0 variable from true(1) to false(2).

The saHpiDiscover.0 variable can also be set to true(1) to initiate a new HPI discovery operation. When the operation completes, the HPI subagent re-populates its tables and changes the value of the saHpiDiscover.0 variable from true(1) to false(2).

Important: SNMP management applications should avoid reading HPI-B0101-MIB module tables while the saHpiDiscover.0 variable has the value true(1), because the HPI subagent is waiting for HPI discovery to complete or is populating its tables.

Initialization in basic mode

The following initialization sequence occurs on ATCA-6002 shelves in basic mode:

1. The *init* script loads the */etc/shmgr_basic.conf* script.
2. The Shelf Manager activates payload power on the module it is running on and then activates all subsidiary FRUs.
3. The Shelf Manager activates the payload power and subsidiary FRUs on the node module in slot 2. It then activates its backplane interfaces.
4. E-Keys are granted to all of the devices that have been detected in the shelf.

Shelf Management Files

This appendix describes files in the shelf management infrastructure, including programs, scripts, and configuration files.

Shelf management configuration files

The configuration files are listed by directory location as follows:

In the */etc* directory:

- *rsyshsd.conf* An optional blade HPI configuration file that overrides the defaults specified in *usr/share/shmgr/shmgr.defs*.
- *shmgr.conf* An optional configuration file that overrides the defaults specified in */usr/share/shmgr/shmgr.defs*.
- *shmgr_basic.conf* An optional configuration file that is used when the Shelf Manager is operating in basic mode on the ATCA-6002 shelf.
- *shmgralarm.conf* An optional configuration file that overrides the default alarm severities.

In the */usr/share/shmgr* directory:

- *shmgr.defs* List of default settings that can be overridden in *shmgr.conf*.

In the */var/lib/shmgr* directory:

- *ShelfFruCopy.bin* This file is the local persistent copy of the valid Shelf FRU Information. It resides on both active and standby SCMs. When the Shelf Manager starts, this file gets created or updated with the Shelf FRU Information identified as good. Changes made to the Shelf FRU Information through the active Shelf Manager are written to this file.
- *ShelfFruFactoryDefault.bin* This file is the factory default version of the Shelf FRU Information and resides on both active and standby SCMs. The file is created the first time the Shelf Manager starts and does not get updated. This file is the only copy of the original Shelf FRU Device image. If the file is deleted, it will get recreated the next time the Shelf Manager is started based on the current Shelf FRU information.
- *atca-6002.bin* This file is the local persistent copy of the shelf FRU data for the ATCA-6002 shelf, which does not have a physical shelf FRU device.

Shelf Management Files

Shelf Manager log file

The Shelf Manager log file is `/tmp/shmgr/shmgr.log`. Backup files in the same directory are created as described in [Configuring verbosity and log settings on page 43](#).

Shelf management executable files

The paths to the following executable files are included in the PATH environment variable:

<i>frurw</i>	Utility for rewriting the contents of an IPMI FRU data device. This utility is required only when upgrading the contents of the Shelf FRU Device and is not required for normal operation.
<i>garp</i>	Utility for generating gratuitous ARP (Address Resolution Protocol) messages over the network. It is invoked by the Shelf Manager during switchover so that the new virtual IP address is recognized correctly by other hosts on the local network.
<i>rsys-ipmitool</i>	Firmware upgrade utility for updating the IPMI firmware and the FPGA code on the IPMC. This utility also enables and disables the Shelf Manager.
<i>keepalive</i>	Tiny program that runs a process in a continuous loop. It will restart the Shelf Manager-HPI server daemon if it stops for some reason.
<i>rmcpping</i>	Utility for issuing RMCP ping messages.
<i>shmgr</i>	Script that starts, stops, restarts, and gets the status of the Shelf Manager. The script calls <i>shmgrd</i> .
<i>shmgrd</i>	Shelf Manager server program.
<i>rsys_pms</i>	Shelf Manager process monitoring service.

IPMI Commands and Managed Sensors

Supported IPMI commands

The Shelf Manager supports all IPMI commands listed in the *AdvancedTCA Base Specification*.

Note:

- On the ATCA-6002 shelf, the Get Telco Alarm Location command does not return a valid location because no telco alarm site exists on this shelf. All other IPMI commands are supported on the ATCA-6002 shelf when the Shelf Manager is running in enhanced mode.
- Do not use the PICMG “FRU control” command with the cold reset option to reset AMCs installed in the ATCA-1200. Doing so will reset the entire ATCA-1200 module. The HPI *saHpiResourceResetStateSet ()* function also uses the PICMG “FRU control” command and will cause the ATCA-1200 module to reset.

Sensors managed by the Shelf Manager

This section lists the logical sensors that are monitored by the Shelf Manager. These include:

- When the shelf has redundant SCMs, the set of logical sensors on each SCM that indicate the status of the active and standby Shelf Managers. See [Table 18 on page 143](#).
- The set of HPI-defined logical sensors for each FRU that indicates the status of the FRU. See [Table 19 on page 144](#).
- The HPI-defined logical sensor for each shelf that indicates the minimum fan speed. See [Table 20 on page 146](#).

Note: These sensors are supported on the ATCA-6002 shelf only when the Shelf Manager is running in enhanced mode.

IPMI Commands and Managed Sensors

Table 18. Managed sensors for the Shelf Manager

#	Name	Type	Category	Normal reading or event state	Explanation of values
4097 (0x1001)	Shelf Manager Redundancy Status	Operational	Redundancy	0x1	State 0x1: Shelf Manager is redundant. State 0x4: Shelf Manager redundancy is degraded due to different Shelf Manager software versions. State 0x8: Shelf Manager is non-redundant due to loss of sufficient resources. State 0x20: Shelf Manager is non-redundant due to insufficient resources.
4098 (0x1002)	Active Shelf Manager Resource	Entity Presence	Presence	0x70082 or 0x80084	Reading represents ResourceId of active Shelf Manager. State 0x1: Resource is absent. State 0x2: Resource is present.
4099 (0x1003)	Standby Shelf Manager Resource	Entity Presence	Presence	0x70082 or 0x80084	Reading represents ResourceId of standby Shelf Manager. State 0x1: Resource is absent. State 0x2: Resource is present.
4111 (0x100F)	Shelf Manager Status	Operational	Sensor-specific	0x1 or 0x2	State 0x1: This module hosts the active Shelf Manager. State 0x2: This module hosts the standby Shelf Manager. State 0x4: It cannot be determined whether this module hosts the active or the standby Shelf Manager.
4199 (0x1067)	Shm Thread Usage	Management Health	Threshold	< 0x50 (< 80%)	Percentage of threads actively used by the Shelf Manager from its thread pool at a point in time. Thresholds: Upper critical: 0x64 (100%) Upper major: 0x50 (80%)
4200 (0x1068)	HSD Thread Usage	Management Health	Threshold	< 0x50 (< 80%)	Percentage of threads actively used by the HPI server from its thread pool at a point in time. Thresholds: Upper critical: 0x64 (100%) Upper major: 0x50 (80%)

IPMI Commands and Managed Sensors

Table 19. Managed sensors for each FRU

#	Name	Type	Category	Normal reading or event state	Explanation of values
256 (0x100)	FRU Operational Status	Operational	Device enabled	0x1	State 0x1: FRU is operational without faults. State 0x2: FRU operation is degraded or disabled. Possible readings are encoded as a bit mask of 64 values:
					Index Meaning
					0 Bused Resource Release
					1 Query for bused resource
					2 Cooling Power Off
					3 Cooling Power On
					4 Shutdown Fan
					5 Get Fan Speed Properties
					6 Get Fan Level (Critical severity)
					7 Get Fan Level (Major severity)
					8 Set Fan Level (Critical severity)
					9 Set Fan Level (Major severity)
					10 Fan Critical Alarm
					11 Self FRU Critical Alarm
					12 Minor Temp Alarm
					13 Major Temp Alarm
					14 Critical Temp Alarm/No Temp Alarms
					15 E-Key load failed
					16 E-Key enable error on <i><interface><channel></i>
					17 Get Port State failed <i><channel></i>
					18 IPMB <i><A/B></i> State
					19 Waiting for system manager to activate
					20 IPM not in M4 or M5, cannot activate
					21 <i><Activation>/<Deactivation></i> Error
					22 Peer LAN sync channel not defined
					23 Local sync channel not started
					24 Cannot calculate peer IP address
					25 Peer LAN channel has unknown properties
					26 Peer LAN session error, closing
					27 Peer LAN connect error
					28 Sync thread exited
					29 Cannot ping peer ShMC

IPMI Commands and Managed Sensors

Table 19. Managed sensors for each FRU (continued)

#	Name	Type	Category	Normal reading or event state	Explanation of values
					30 Get Address Info failed
					31 Self Test Error
					32 Self Test Check failed
					33 Set IPMB-<A/B> State <Enable/Disable>
					34 Comm Loss (M7)
					35 <System Interface> Status
					36 Ignoring SDR <sdr> with invalid owner <name/id>
					37 Can't read DLR
					38 Can't read SDR<record id>
					39 Could not get DPT info
					40 Could not reserve DPT
					41 Could not get DPT entry<entry id>
					42 Stable Redundancy Role
					43 Local HA State Event Lost
					44 Peer HA State Event Lost
					45 Cannot ping local ShMC
257 (0x101)	FRU Power Status	Power Unit	Threshold	0x1	State 0x1: Lower minor. State 0x2: Lower major. State 0x4: Lower critical. State 0x8: Upper minor. State 0x10: Upper major. State 0x20: Upper critical. Possible readings are encoded as a bit mask of 64 values:
					0 No power information record, cannot activate
					1 Get Power Level failed, assuming worst case
					2 No Shelf FRU power descriptor found
					3 No Shelf FRU power feeds attached
					4 Get Power Level (Level Type) failed
					5 Number of early/desired levels don't match
					6 Number of desired levels is zero
					7 Power level zero desired
					8 Early draw is less than steady draw
					9 Set Power Level failed

IPMI Commands and Managed Sensors

Table 19. Managed sensors for each FRU (continued)

#	Name	Type	Category	Normal reading or event state	Explanation of values
258 (0x102)	FRU Thermal Status	Temperature	Threshold	0x1	State 0x1: Lower minor. State 0x2: Lower major. State 0x4: Lower critical. State 0x8: Upper minor. State 0x10: Upper major. State 0x20: Upper critical. Possible readings are encoded as a bit mask of 64 values:
					0 Minor Temp Alarm
					1 Major Temp Alarm
					2 Critical Temp Alarm

Table 20. Managed sensor for each shelf

#	Name	Type	Category	Normal reading or event state	Explanation of values
4198 (0x1066)	Shelf Fan Floor Level	Cooling Device	Sensor-Specific	0x1E (30%)	The current fan floor level in hex as a percentage of the maximum fan speed.